



(51) 国際特許分類6
G09C 1/00, H04L 9/06

A1

(11) 国際公開番号

WO99/38143

(43) 国際公開日

1999年7月29日 (29.07.99)

(21) 国際出願番号 PCT/JP99/00337
(22) 国際出願日 1999年1月27日 (27.01.99)
(30) 優先権データ
特願平10/13572 1998年1月27日 (27.01.98) JP
特願平10/13573 1998年1月27日 (27.01.98) JP
特願平10/147479 1998年5月28日 (28.05.98) JP
(71) 出願人 (米国を除くすべての指定国について)
日本電信電話株式会社(NIPPON TELEGRAPH AND TELEPHONE CORPORATION)[JP/JP]
〒163-8019 東京都新宿区西新宿3丁目19番2号 Tokyo, (JP)
(72) 発明者 ; および
(75) 発明者／出願人 (米国についてのみ)
仲田雅透(KANDA, Masayuki)[JP/JP]
島嶋洋一(TAKASHIMA, Youichi)[JP/JP]
青木和麻呂(AOKI, Kazumaro)[JP/JP]
道田広樹(UEDA, Hiroki)[JP/JP]
太田和夫(OHTA, Kazuo)[JP/JP]
〒163-1419 東京都新宿区西新宿3丁目20-2
日本電信電話株式会社内 Tokyo, (JP)

松本 勉(MATSUMOTO, Tsutomu)[JP/JP]
〒227-0048 神奈川県横浜市青葉区柿の木台13-45
Kanagawa, (JP)
(74) 代理人
弁理士 草野 卓, 外(KUSANO, Takashi et al.)
〒160-0022 東京都新宿区新宿四丁目2番21号
相模ビル Tokyo, (JP)

(81) 指定国 CA, US, 欧州特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE)

添付公開書類
国際調査報告書

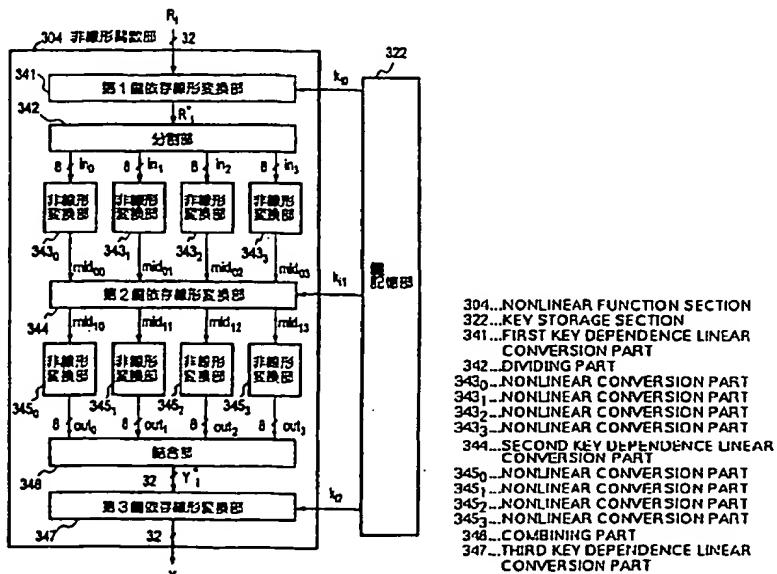
54) Title: DATA CONVERTER AND RECORDING MEDIUM ON WHICH PROGRAM FOR EXECUTING DATA CONVERSION IS RECORDED

54) 発明の名称 データ変換装置及びそれを実施するプログラムが記録された記録媒体

57) Abstract

A data converter has rounding units (38) each including a nonlinear function section 304, and each nonlinear function section (304) includes a first key dependence linear conversion part (341) for performing linear conversion based on a subkey, a dividing part (342) for dividing the output of the first key dependence linear conversion part (341) into n pieces of sub-data, first nonlinear conversion parts (343) for nonlinearly converting the sub-data, a second key dependence linear conversion part (344) for linearly converting the nonlinear conversion outputs based on a subkey and outputting n pieces of conversion sub-data, second nonlinear conversion parts (345) for nonlinearly converting the conversion sub-data, and a combining part 346 for combining the nonlinear conversion outputs and producing the output of the nonlinear function section. An nxn matrix representing the linear conversion by the second key dependence linear conversion part (344) is composed of n vectors such that the hamming weight for a safety threshold T is T-1 or more.

Therefore, the resistance to the difference decryption method and the linear decryption method is strengthened.



(57)要約

それぞれ非線形関数部(304)を含む複数のラウンド処理部(38)を有し、各非線形関数部(304)は、副鍵に基づいて線形変換を行う第1鍵依存線形変換部(341)と、その出力をn個のサブデータに分割する分割部(342)と、それらサブデータをそれぞれ非線形変換する第1非線形変換部(343)と、それらの非線形変換出力を副鍵に基づいて線形変換し、n個の変換サブデータを出力する第2鍵依存線形変換部(344)と、それら変換サブデータを非線形変換する第2非線形変換部(345)と、その非線形変換出力を結合して非線形関数部の出力とする結合部(346)を含み、その第2鍵依存線形変換部(344)の線形変換処理を表すn×n行列を、安全性閾値Tに対しハミング重みがT-1以上となるn個のベクトルにより構成することにより差分解読法、線形解読法に対し耐性を強くする。

PCTに基づいて公開される国際出願のパンフレット第一頁に掲載されたPCT加盟国を同定するために使用されるコード(参考情報)

AE アラブ首長国連邦	ES スペイン	LI リヒテンシュタイン	SG シンガポール
AL アルバニア	FI フィンランド	LK スリ・ランカ	SI スロヴェニア
AM アルメニア	FR フランス	LR リベリア	SK スロヴァキア
AT オーストリア	GA ガボン	LS レントア	SL シエラ・レオネ
AU オーストラリア	GB 英国	LT リトアニア	SN セネガル
AZ アゼルバイジャン	GD グレナダ	LU ルクセンブルグ	SZ スウェーデン
BA ボスニア・ヘルツェゴビナ	GE グルジア	LV ラトヴィア	TD ティード
BB バルバドス	GH ガーナ	MC モナコ	TG トーゴ
BE ベルギー	GM ガンビア	MD モルドヴァ	TJ タジキスタン
BF ブルガリア	GN ギニア	MG マダガスカル	TM トルクメニスタン
BC ブルガリア	GW ギニア・ビサオ	MK マケドニア IE ユーゴスラヴィア	TR トルコ
BJ ベナン	GR ギリシャ	共和国	TT トリニダード・トバゴ
BR ブラジル	HR クロアチア	ML マリ	UA ウクライナ
BY ベラルーシ	HU ハンガリー	MN モンゴル	UG ウガンダ
CA カナダ	ID インドネシア	MR モーリタニア	US 米国
CF 中央アフリカ	IE アイルランド	MW マラウイ	UZ ウズベキスタン
CG コンゴ	IL イスラエル	MX メキシコ	VN ヴィエトナム
CH スイス	IN インド	NE ニジエール	YU ユーロッパ
CI ニートジボアール	IS アイスランド	NL オランダ	ZA 南アフリカ共和国
CM カメルーン	IT イタリア	NO ノールウェー	ZW ジンバブエ
CN 中国	JP 日本	NZ ニュージーランド	
CU キューバ	KE ケニア	PL ポーランド	
CY キプロス	KG キルギスタン	PT ポルトガル	
CZ チェコ	KP 北朝鮮	RO ルーマニア	
DE ドイツ	KR 韓国	RU ロシア	
DK デンマーク	KZ カザフスタン	SD スーダン	
EE エストニア	LC セントルシア	SE スウェーデン	

明細書

データ変換装置及びそれを実施するプログラムが記録された記録媒体

技術分野

この発明は、データの通信または蓄積において、データを秘匿するための暗号化装置に用いられるデータ変換装置、特に、秘密鍵の制御のもとでデータをブロック単位で暗号化または復号を行う共通鍵暗号方式による暗号化装置に適するデータ変換装置及びそのデータ変換装置により実行するプログラムを記録した記録媒体に関する。

従来の技術

高速かつ安全な共通鍵暗号を構成するために、暗号化対象のデータを適當な長さのブロックに分割し、そのデータブロック毎に暗号化する方法をブロック暗号と呼ぶ。通常、ブロック暗号は、暗号化の対象である入力データを攪乱するためのデータ拡散部と、暗号装置に入力された秘密の共通鍵（以降ではこの鍵を主鍵と呼ぶ）を入力としてデータ拡散部が利用する一連の副鍵を生成するための鍵スケジュール部とから構成されている。データを秘匿するためのこのようなデータ変換装置に使用される代表的な共通鍵暗号方式には、米国連邦標準暗号であるDES (Data Encryption Standard) 暗号がある。

図1は、DES暗号の機能構成を示す。DES暗号では、64ビットの秘密鍵（うち8ビットはパリティに用いられる）を用い、64ビットのデータブロック単位に暗号化または復号を行う。図1において、暗号化処理は、データ拡散部10において平文Mの64ビットを初期転置部11において初期転置で変換した後、32ビットごとのブロックデータL₀、R₀に分割される。次に、ブロックデータR₀は図2の第iラウンド処理部14_i (i=0, 1, ..., 15) に示すデータ変換部である関数演算部（ラウンド関数とも呼ばれる）12へ入力され、48ビットの副鍵k₀の制御のもとにf(R₀, k₀)に変換される。この変換データf(R₀, k₀)とブロックデータL₀との排他的論理和をXOR回路13でとり、更に、その出力値とブロックデータR₀とを入れ替えて、次のブロックデータL₁、R₁とする。即ち、

$$R_1 = L_0 \oplus f(R_0, k_0)$$

$$L_1 = R_0$$

である。ここで \oplus は排他的論理和を表わすものとする。このように2つのブロックデータ L_0, R_0 を入力として演算部12と排他的論理和回路13とデータの入れ替え(スワップ)とにより L_1, R_1 を出力する第0段ラウンド処理部14₀が構成され、同じようなラウンド処理部14₁～14₁₅が縦続的に設けられる。第i段ラウンド処理部14_iによる処理を第i段のラウンド処理と呼ぶことにする。ただし、 $i=0, 1, \dots, 15$ である。つまり各ラウンド処理部14_i ($0 \leq i \leq 15$) では、

$$R_{i+1} = L_i \oplus f(R_i, k_i)$$

$$L_{i+1} = R_i$$

の処理が行われ、最後に R_{16}, L_{16} を統合して64ビットにした後、最終転置部15において最終転置で変換して暗号文64ビットを出力する。なお、最終転置部15は初期転置部11の逆変換に相当する。

復号処理においては、関数f(関数演算部12)に入力する副鍵 k_0, k_1, \dots, k_{15} の順序だけを逆転させて、 $k_{15}, k_{14}, \dots, k_1, k_0$ の順に入力する点を除けば、暗号化処理と同じ手順で実行できる。その場合、最終段ラウンド処理部14₁₅のスワップ出力 L_{16}, R_{16} を、図に示すように更にスワップするように構成することにより、復号処理において暗号文を初期転置11に入力して図1の処理を実行することにより、最終転置15の出力に平文がそのまま得られる。鍵スケジュール部20は、拡大鍵生成ルーチン16で64ビットの主鍵から8ビット分のパリティビットを除いた56ビットを左右28ビットずつの鍵データに分割し、それら各28ビットの左右鍵データに対し16段の並び替え処理を行い、各段の並び替え処理結果の左右鍵データ(計56ビット)を縮約転置によりデータ拡散部10の対応するラウンドに与える16個の48ビットの副鍵 k_0, k_1, \dots, k_{15} として生成する。

関数演算内部12の処理は、図2に示すように行われる。まず、32ビットのブロックデータ R_i は拡大転置部17で48ビットデータ $E(R_i)$ に変換される。これに副鍵 k_i とで排他的論理和をXOR回路18で取り、48ビットデータ $E(R_i) \oplus k_i$ に変換した後、8個の6ビットごとのサブブロックデータに分割する。この8個

のサブブロックデータはそれぞれ異なるS-box $S_0 \sim S_7$ に入力され、各々が4ビットの出力を得る。なお、このS-box S_j ($j=0, 1, \dots, 7$) は6ビットの入力データから4ビットの出力データに変換する非線形変換テーブルであり、DES暗号の本質的な安全性を担っている部分である。S-box $S_0 \sim S_7$ の8つの出力データは、再び連結されて32ビットデータになった後、転置変換部19を経て、図2に示されるように関数演算部12の出力 $f(R_i, k_i)$ となる。この出力は、図1に示されるように、 L_i と排他的論理和されて R_{i+1} となる。

次に、暗号解読法について述べる。DES暗号を始めとする従来の共通鍵暗号方式についてはさまざまな方面から暗号解読が試みられており、そのなかでも、極めて効果的な解読法は、E. Biham およびA. Shamirによって“Differential Cryptanalysis of DES-like Cryptosystems,” Journal of Cryptology, Vol. 4, No. 1, pp. 3-72に提案された差分解読法と、松井によって“Linear Cryptanalysis Method for DES cipher,” Advances in Cryptology-EUROCRYPT'93 (Lecture Notes in Computer Science 765), pp. 386-397に提案された線形解読法である。

DES暗号に対する差分解読法は、2つのデータ X, X^* の差分を

$$\Delta X = X \oplus X^*$$

としたとき、解読者が入手している平文・暗号文の2組を以下の式に適用して、最終ラウンド14₁₅における副鍵 k_{15} を求めることを目的としている。図1の暗号化処理において、第1及び第2平文を入力したときの各ラウンド処理部14_iでの入力ブロックデータを $(L_i, R_i), (L^*_i, R^*_i)$ とする。上記差分の定義により次式

$$\Delta L_i = L_i \oplus L^*_i$$

$$\Delta R_i = R_i \oplus R^*_i$$

が成立する。図1において、 $L_{15} = R_{14}$, $L^*_{15} = R^*_{14}$, $L_{16} = R_{15}$, $L^*_{16} = R^*_{15}$ なので次式

$$R_{16} = L_{15} \oplus f(R_{15}, k_{15})$$

$$R^*_{16} = L^*_{15} \oplus f(R^*_{15}, k_{15})$$

が成立し、これら2つの式の両辺の排他的論理和をとると次式

$$\Delta R_{16} = \Delta L_{15} \oplus f(L_{15}, k_{15}) \oplus f(L_{16} \oplus \Delta L_{15}, k_{15})$$

が得られ、その両辺と $\Delta R_{14} = \Delta L_{15}$ の排他的論理和をとることにより次の式が得ら

れる：

$$f(L_{16}, k_{15}) \oplus f(L_{16} \oplus \Delta L_{16}, k_{15}) = \Delta R_{16} \oplus \Delta R_{11}$$

このとき、 L_{16} 、 ΔL_{16} 、 ΔR_{16} は暗号文から得られるデータであるので入手済みの情報である。このため、解読者が ΔR_{11} を正しく求めることができるならば、上式は k_{15} のみが未知定数となり、入手済みの平文・暗号文の組を用いて k_{15} に関する全数探索を行うことで、解読者は必ず正しい k_{15} を見つけだすことができる。従って、副鍵 k_{15} が求まってしまえば、残り 8 (即ち 56-48) ビットは総当たりでも簡単に求まってしまう。

一方、 ΔR_{11} についてみてみると、この値は中間差分値であるため、一般には求めることが困難である。そこで、第 0 ラウンドから最終ラウンドの一つ前までの第 1 4 段ラウンドまでにおいて、各ラウンド (段) が確率 p_i で次式

$$\Delta R_{i+1} = \Delta L_i \oplus \Delta \{f(\Delta R_i)\}$$

$$\Delta L_{i+1} = \Delta R_{i+1}$$

のように近似されるとおく。ここでのポイントは、ある ΔR_i が入力されたとき、副鍵 k_i の値に関わらず、確率 p_i で $\Delta \{f(\Delta R_i)\}$ を予測できるということにある。このように近似できるのは、非線形な変換である S-box において、入力差分によつては差分出力の分布に極めて大きな偏りが生じるためである。例えば、S-box S_0 では、入力差分 " $110100_{(2)}$ " のとき、 $1/4$ の確率で出力差分 " $0010_{(2)}$ " に変換されるためである。そこで、各々の S-box が確率 p_{s_i} で入力差分と出力差分との関係が予測できるとおき、これらを組み合わせることで各ラウンドの近似を求める。更に、各ラウンドでの近似を連結していくことで、 ΔR_{11} は確率 $P = \prod_{i=0}^{13} p_i$ で ΔL_0 、 ΔR_0 (ΔL_0 、 ΔR_0 は平文から得られるデータであるので入手済みの情報である) から求められることになる。なお、この確率 P が大きいほど、暗号解読が容易である。このようにして、副鍵 k_{15} が求められると、今度は 1 段少ない 15 段 DES 暗号とみなして、同様の手法で、副鍵 k_{11} を求めていくということを繰り返して、最終的に副鍵 k_0 まで求めていく。

この解読法による暗号解読が成功するかどうかは確率 P に依存し、この値が大きいほど成功しやすい。Biham らによると、この解読法では、 2^{17} 組の解読者が選

択した既知平文・暗号文の組を入手できればDES暗号を解読できるとしている。

また、DES暗号に対する線形解読法は、以下の線形近似式を構成し、解読者が入手している平文・暗号文の組による最尤法を用いて副鍵を求める目的としている。

$$\begin{aligned} & (L_0, R_0) \Gamma (L_0, R_0) \oplus (L_{16}, R_{16}) \Gamma (L_{16}, R_{16}) \\ & = (k_0, k_1, \dots, k_{15}) \Gamma (k_0, k_1, \dots, k_{15}) \end{aligned}$$

ただし、 $\Gamma(X)$ は X の特定のビット位置を選択するベクトルを表し、マスク値という。

線形近似式の役割は、暗号アルゴリズム内部を線形表現で近似的に置き換え、平文・暗号文の組に関する部分と副鍵に関する部分とに分離することにある。つまり、平文・暗号文の組に関して、平文の特定のビット位置の値と暗号文の特定のビット位置の値との全ての排他的論理和が一定値となり、その値は副鍵の特定のビット位置の値の排他的論理和に等しくなることを表している。従って、解読者は

$$(L_0, R_0) \Gamma (L_0, R_0) \oplus (L_{16}, R_{16}) \Gamma (L_{16}, R_{16})$$

の情報から

$$(k_0, k_1, \dots, k_{15}) \Gamma (k_0, k_1, \dots, k_{15}) \quad (1 \text{ ビット})$$

の情報が得られるということになる。このとき、 (L_0, R_0) 、 (L_{16}, R_{16}) はそれぞれ平文・暗号文のデータであるので入手済みの情報である。このため、解読者が $\Gamma(L_0, R_0)$ 、 $\Gamma(L_{16}, R_{16})$ 、 $\Gamma(k_0, k_1, \dots, k_{15})$ を正しく求めることができれば、 $(k_0, k_1, \dots, k_{15}) \Gamma (k_0, k_1, \dots, k_{15}) \quad (1 \text{ ビット})$ を求めることができる。

DES暗号では、非線形な変換が起きる部分はS-box しかないため、S-boxについてのみ線形表現ができれば、容易に線形近似式が構成できる。そこで、各々のS-box が確率 $p_{s,i}$ で線形表現できるとおく。ここでのポイントは、S-box に対する入力マスク値が与えられたとき、確率 $p_{s,i}$ でその出力マスク値を予測できるということにある。これは、非線形変換テーブルであるS-box において、入力マスク値によっては差分マスク値の分布に極めて大きな偏りが生じるためにおこる。例えれば、S-box S_4 では、入力マスク値 "010000₍₂₎" のとき、3/16の確率で出力マスク値

"1111₍₂₎"が予測されるためである。これらS-boxにおけるマスク値を組み合わせることによって、各ラウンドが確率 p_i で入力マスク値と出力マスク値との線形表現ができ、各ラウンドでの線形表現を連結していくことで、 $\Gamma(L_0, R_0)$, $\Gamma(L_{15}, R_{15})$, $\Gamma(k_0, k_1, \dots, k_{15})$ は確率

$$P = 1/2 + 2^{15} \prod_{i=0}^{15} |p_i - 1/2|$$

で求められることになる。なお、この確率 P が大きいほど、暗号解読が容易である。

松井によると、この解読法で、 2^{43} 組の既知平文・暗号文の組を用いて、DES暗号の解読に成功している。

さて、上記の解読法に対抗するためには、確率 P が十分に小さくなればよい。このため、確率 P を小さくするための提案がさまざま行われており、なかでも従来の暗号方式において、もっとも簡単に安全性を高めるための方法がラウンド数（段数）を増やすことであった。例えば、DES暗号を3つつなげたTriple-DES暗号は、実質的にDESのラウンド数を16段から48段に増やした暗号方式であり、確率 P は、DES暗号よりもはるかに小さい。

しかし、上記の解読法に対抗するための対策として、ラウンド数（段数）を増加させることは、暗号化速度を犠牲にすることになる。例えば、ラウンド数を3倍に増やせば、暗号化速度は $1/3$ になる。つまり、現在のDES暗号の暗号化速度はPentium PCクラスで約10Mbpsであるため、Triple-DES暗号ともなると約3.5Mbpsまで暗号化速度が低下する。一方で、ネットワークやコンピュータなどは年々高速化しており、データ変換装置もそれらの高速化に対応したものが望まれている。このため、従来のデータ変換装置では、それらの高速化の要求に対して、安全性と高速性を同時に満たすことはきわめて困難な状況になっている。

また、上述のように、差分解読法や線形解読法では最終ラウンドの副鍵が求められる。DES暗号の場合、最終ラウンドの副鍵が求められてしまうと、そこから容易に主鍵が求められてしまうという欠点があるため、鍵スケジュール部20の副鍵と主鍵の対応関係を更に複雑にすることで、安全性を向上する方法が米国特許No.4,850,019に提案されている。その原理的構成を図3に示す。上記米国特

許では、データ拡散部(f_k)を用いて主鍵から副鍵を生成することで、例えばデータ拡散部(f_k)を用いて主鍵から副鍵を生成することで、例えば副鍵が求まったとしても簡単には主鍵が求まらないことを期待している。

上記米国特許で示されている鍵スケジュール部20の概要を図3を参照して次に説明する。拡大鍵生成ルーチン21はそれぞれ鍵拡散部 $22_0 \sim 22_{N/2-1}$ を有する $N/2$ (例えば $N=16$)段の鍵処理部 $21_0 \sim 21_{N/2-1}$ から成る。各鍵処理部 21_j ($j=0, 1, \dots, N/2-1$)は与えられた各32ビットの入力左右鍵データを拡散処理し、その出力左右鍵データは左右が入れ替わられて次段の鍵処理部 21_{j+1} に左右鍵データとして入力される。初段を除く各鍵処理部 21_j は排他的論理和部 23_j を有し、前段の鍵処理部 21_{j-1} の入力左鍵データと出力左データの排他的論理和を演算し、その演算結果を鍵拡散部 22_j に与える。鍵処理部 21_j の入力左鍵データは鍵拡散部 22_j で排他的論理和 23_j の出力により拡散され次段の入力右鍵データとして出力され、鍵処理部 21_j の入力右鍵データは次段の入力左鍵データとして出力される。各鍵拡散部 22_j の出力はビット分割されて2つの副鍵 Q_{2j}, Q_{2j+1} (即ち k_i, k_{i+1})として図1の対応する第 $i=2j$ ラウンド処理段と第 $i+1=2j+1$ ラウンド処理段に与えられる。

64ビットの主鍵はそれぞれ32ビットの左右鍵データに分割され、初段の鍵処理部 21_0 において左鍵データを右鍵データで鍵拡散部 22_0 により拡散して拡散左鍵データとし、この拡散左鍵データと右鍵データの左右を入れ替えて左右鍵データとして次段の鍵処理部 21_1 に与える。鍵処理部 $21_0 \sim 21_{N/2-1}$ の鍵拡散部 $22_0 \sim 22_{N/2-1}$ の出力は副鍵 $k_0 \sim k_{N-1}$ として図1に示したデータ拡散部10の対応するラウンド $14_0 \sim 14_{N-1}$ に与えられる。

しかしながら、図3の拡大鍵生成ルーチン21において、各鍵拡散部 22_j は2つの入力データから1対の鍵データ(副鍵 Q_{2j}, Q_{2j+1})を生成出力する関数であり、これら2つの入力データの一方と、出力データとが判明すれば、他方の入力データを知ることができる性質を備えているとき、仮に3対の副鍵 $Q_{2j-2}, Q_{2j-1}; Q_{2j}, Q_{2j+1}; Q_{2j+2}, Q_{2j+3}$ が判明したと仮定すると、第 $j+1$ 段の鍵拡散部 22_{j+1} において、その出力(副鍵 Q_{2j+2}, Q_{2j+3})と、一方の入力データ(副鍵 Q_{2j-2}, Q_{2j-1})がわかっているので、他の入力データ(即ち排他的論理和部 23_{j+1} の出力データ)を求めることができ、その

求めたデータと排他的論理和部23_{j+1}の一方の入力データである副鍵Q_{2j}, Q_{2j+1} とか
ら排他的論理和部23_{j+1}の他方の入力データである前段（第 j 段）の鍵拡散部22_jの
入力、即ち、3 段前（第 j-2 段）の鍵拡散部22_{j-2}の出力である副鍵Q_{2j-1}, Q_{2j-3} が求
まる。この様な操作を順次繰り返すことで、データ拡散部 10 におけるデータを
解析すること無しに、鍵スケジュール部 20 におけるデータのみを解析すること
で全ての副鍵を決定できてしまう。ここでは連続した 3 段分の副鍵が求まると全
ての副鍵が求まることを指摘したが、連続 2 段分の副鍵を知っている場合には、
隣接する他の 1 段分の副鍵を総当たりで推定することでも、攻撃に成功する。

図1のラウンド処理の最終段を $i=N$ とすると、差分解読法及び線形解読法では、副鍵 k_s, k_{s-1} が求まりやすい。それらを使って上述のように鍵スケジュール部21における鍵データの解析を行うことにより全ての副鍵が求まる可能性がある。

この発明の第1の目的は、ラウンド関数 f (関数演算部) を安全性と高速性を同時に満たすような構造にすることによって、ラウンド数 (段数) を大幅に増加させることなく安全性を確保し、かつ高速な暗号化処理が可能となるようなデータ変換装置及びデータ変換を実施するプログラムを記録した記録媒体を提供することにある。

この発明の第2の目的は、一部の副鍵が知られた場合においても、鍵スケジュール部を解析するだけでは他の副鍵及び主鍵が簡単には求まらないような鍵スケジュール部を実現することである。

発明の開示

この発明の第1の目的を達成するため、特に非線形関数部において、非線形関数部の入力データに対し鍵記憶部に蓄積された第1鍵データに基づいて線形変換を行う第1鍵依存線形変換部と、その第1鍵依存線形部の出力データをn個のサブデータに分割する分割部と、これらの各サブデータに非線形変換を行う第1非線形変換部と、その第1非線形変換部の各々の出力サブデータに対し第2鍵データに基づいて線形変換を行う第2鍵依存線形変換部と、その第2鍵依存線形変換部の出力サブデータに非線形変換を行う第2非線形変換部と、その第2非線形変換部の出力サブデータを結合して非線形関数部の出力データとする結合部とを

備えており、上記第2鍵依存線形変換手段は、その入力に対し $n \times n$ 行列で規定される排他的論理和を行う線形変換部を含んでいることを特徴とする。

この発明によれば、第1及び第2非線形変換部における差分確率・線形確率が p (< 1) であるとき、各段を近似するときの差分確率・線形確率は $p_i \leq p^2$ (ただし、差分解読法では関数 f (非線形関数部) への入力差分が 0 でないとき、線形解読法では関数 f での出力マスク値が 0 でないとき) となることが保証される。また、関数 f が全単射であるとき、暗号装置の段数を $3r$ とすると、暗号装置としての確率は $P \leq p_i^{2r} \leq p^{4r}$ となる。更に、特に $n = 4$ の場合の第2鍵依存線形変換部が 4 つのサブデータから選んだ 3 つと鍵データの 4 分割の 1 つの排他的論理和の全ての組合せを演算する構造を有するならば、各段を近似するときの確率は $p_i \leq p^4$ 、暗号装置としての確率は $P \leq p_i^{2r} \leq p^{8r}$ となる。 $n = 8$ の場合の第2鍵依存線形変換部が 8 つのサブデータから選んだ 6 つ又は 5 つと、鍵データの 8 分割の 1 つの排他的論理和の全ての組み合わせを演算する構造を有するならば、格段を近似するときの確率は $p_i \leq p^5$ となり、暗号装置としての確率は $P < p_i^{2r} < p^{10r}$ となる。

また、第1および第2非線形変換部はそれぞれの手段が完全に並列処理できるような配置となっているため、高速化に寄与する。

ゆえに、差分解読法や線形解読法に対する安全性を有した高速な非線形関数を構成でき、安全性と高速性を両立させたデータ変換装置を提供することが可能になる。

この発明の第2の目的を達成するため、鍵スケジュール装置において、鍵拡散部 (関数 f) と同様の働きをする G 関数部を用い、G 関数部の出力である L 成分が記憶部に一旦記憶され、必要な個数だけ L 成分を求めた後に、それぞれの L 成分から出来るだけ均一に必要となる情報を抽出して副鍵を生成するデータ抽出機能を備えた H 関数部が設けられる。更に、G 関数部の出力である L 成分からそれぞれの副鍵として使用される部分情報が H 関数部で抽出され、記憶部に記憶され、必要な個数の L 成分から部分情報を抽出することで副鍵が生成される。

図面の簡単な説明

図 1 は従来の D E S 暗号装置の機能構成を示す図。

図 2 は図 1 中の f 関数演算部 1 2 の具体的機能構成を示す図。

図 3 は図 2 における拡大鍵生成ルーチン 21 の構成例を示す図。

図 4 はこの発明の第 1 実施例の機能構成を示す図。

図 5 は第 1 実施例における非線形関数部 304 の詳細な機能構成例を示す図。

図 6 は図 5 における最適線形変換部を決めるための非線形関数部の基本構成図。

図 7 は図 5 中の第 2 鍵依存線形変換部 347 の具体例を示す図。

図 8 A は第 2 実施例における非線形変換部 343 の等価的機能構成を示す図。

図 8 B は第 2 実施例における非線形変換部 344 の等価的機能構成を示す図。

図 8 C は第 2 実施例における非線形変換部 345 の等価的機能構成を示す図。

図 8 D は第 2 実施例における非線形変換部 346 の等価的機能構成を示す図。

図 9 はこの発明の第 2 実施例における第 2 鍵依存線形変換部 347 の機能構成を示す図。

図 10 は実施例 3 における非線形関数部 343 の機能構成を示す図。

図 11 はデータ変換処理をコンピュータで実施する処理手順を示すフローチャート。

図 12 は図 11 におけるステップ S 3 の詳細な処理手順を示すフローチャート。

図 13 はこの発明の第 4 実施例の機能構成を示す図。

図 14 は図 13 における非線形関数部 304 の機能構成を示す図。

図 15 A は探索演算を削減するため限定した構成の線形変換部を示す図。

図 15 B は図 15 A における 1 つの変換ボックスの構成を示す図。

図 16 は探索アルゴリズムにより決定した線形変換部 344A の構成例を示す図。

図 17 は第 4 実施例における図 14 中の第 2 鍵依存線形変換部 344 の機能構成例を示す図。

図 18 は第 4 実施例における図 14 中の第 2 鍵依存線形変換部 344 の他の機能構成を示す図。

図 19 は第 4 実施例における図 14 中の第 2 鍵依存線形変換部 344 の更に他の機能構成を示す図。

図20Aは第5実施例における非線形変換部343_n'の機能構成を示す図。

図20Bは非線形変換部343₁'の機能構成を示す図。

図20Cは非線形変換部343₁'の機能構成を示す図。

図21は第5実施例における第2鍵依存線形変換部344の機能構成を示す図。

図22は記録媒体に記録されたデータ処理手順プログラムを実行する構成を示す図。

図23Aはこの発明による鍵生成スケジュールの原理機能構成を示すブロック図。

図23Bはこの発明による他の鍵生成スケジュールの原理機能構成を示すブロック図。

図24は図23A又は23B中の中間鍵生成部230の機能構成例を示すブロック図。

図25はこの発明を図3の鍵スケジュール部に適用した場合の図24中のG関数部22の機能的構成を示すブロック図。

図26はこの発明を図3の鍵スケジュール部に適用した場合の図23A中の副鍵生成部240の機能的構成を示すブロック図。

図27はこの発明を図3の鍵スケジュール部に適用した場合の図23B中の副鍵生成部250の機能的構成例を示すブロック図（この実施例では副鍵生成部がビット抽出機能をそなえたH関数部をその一部に含む）

図28はこの発明を128ビットを1つのブロックとするFeistel型暗号に適用できるように応用した場合のG関数部22の機能的構成を示すブロック図。

発明を実施する最良の形態

第1実施例

以下、この発明の一実施例を図面を用いて説明する。

図4は、この発明の一実施例を示すデータ変換装置における、暗号化処理手順の機能構成を示したものである。データ変換装置は、データ拡散部10と鍵スケジュール部20とから構成され、この発明によるデータ変換装置においても、デ

ータ拡散部10には入力データを左、右ブロックデータ L_0, R_0 に分割し、それらを順次ラウンド処理するN段に継続接続されたラウンド処理部38₀～38_{N-1}が設けられ、各ラウンド処理部38_i ($i=0, 1, \dots, N-1$) は図1のラウンド関数部12に対応する非線形関数部304と、図1のXOR回路13に対応する線形演算部305と、交換部306とから構成されている。

平文に相当する入力データMを入力部301から暗号装置内に入力する。鍵スケジュール部20は、鍵入力部320と、鍵データ生成部321と、鍵記憶部322とかから構成され、予め、鍵入力部320から入力されたデータ（主鍵K）に基づいて鍵データ生成部321により複数の鍵データ（副鍵）

$$\{fk; k_{00}, k_{01}, k_{02}; k_{10}, k_{11}, k_{12}; \dots; k_{(N-1)0}, k_{(N-1)1}, k_{(N-1)2}; ek\}$$

が生成され、鍵記憶部322に保持される。入力平文データMは、鍵記憶部322に蓄積されている鍵データfkによる初期鍵依存変換部302で変換された後、初期分割部303で左、右ブロックデータ L_0, R_0 に分割される。例えば64ビットのデータが32ビットずつのブロックデータ L_0, R_0 にビット分割される。初期鍵依存変換部302では、例えば鍵データfkと入力データMとの排他的論理和や鍵データfkによる入力データMのビットローテーション（ビット回転）などの線形変換、もしくは乗算などを組み合わせた非線形変換が行われる。

右ブロックデータ R_0 は、鍵記憶部322に蓄積されている鍵データ k_{00}, k_{01}, k_{02} とともにこの発明により特徴的に構成された非線形関数部304に入力され、非線形関数部304で非線形変換処理によりデータ Y_0 に変換される。データ Y_0 と左ブロックデータ L_0 は線形演算部305で線形演算されてデータ L_0^* に変換される。データ L_0^* とデータ R_0 は交換部306でデータ位置の交換（スワップ）が行われ、 $L_1 \leftarrow R_0, R_1 \leftarrow L_0^*$ とされ、 L_1, R_1 が次の第1段ラウンド処理部38₁に入力される。

以下、第i段ラウンド処理部38_i ($i=0, 1, \dots, N-1$)において、2つの入力ブロックデータ L_i, R_i について上記と同様の処理を繰り返し行う。即ち、左、右ブロックデータ L_i, R_i について、データ R_i は、鍵記憶部322に蓄積されている鍵データ k_{i0}, k_{i1}, k_{i2} とともに非線形関数部304に入力され、非線形関数部304で非線形変換処理が行われて、データ Y_i に変換される。データ Y_i とデータ L_i は線形演算部305で演算さ

れてデータ L_i^* に変換される。データ L_i^* とデータ R_i は交換部306 でデータ位置の交換が行われ、 $L_{i-1} \leftarrow R_i$, $R_{i-1} \leftarrow L_i^*$ のように交換される。線形演算部305 は例えば排他的論理和演算を行うものである。

暗号化を行うためのデータ変換装置としての安全性を確保するための適切な繰り返し回数（ラウンド回数）をN とすると、ラウンド処理部38₀～38_{N-1} による繰り返し処理の結果、左、右ブロックデータ L_i , R_i が得られる。このデータ L_i , R_i を最終結合部307 で結合し、つまり例えば32ビットの各 L_i , R_i をビット結合して64ビットのデータとし、その後、鍵記憶部322 に蓄積されている鍵データ ek による最終鍵依存変換部308 で変換し、出力部309 から暗号文として出力データ C を出力する。

復号については、暗号化処理手順と逆の手順をたどることによって、暗号文 C から平文 M が得られる。特に、最終鍵依存変換部308 が、初期鍵依存変換部302 の逆変換になっているならば、図4において入力データの代りに暗号文データを入力し、鍵データを図4とは逆に、 ek , $k_{(N-1)0}$, $k_{(N-1)1}$, $k_{(N-1)2}$, …, k_{10} , k_{11} , k_{12} , k_0 , k_{01} , k_{02} , fk を順次与えればよい。

次に、非線形関数部304 の内部を詳細に説明する。図5は、非線形関数部304 の内部の機能構成を抜き出して示したものである。

第i段ラウンド処理部38_i の入力ブロックデータ R_i は、鍵記憶部322 に蓄積されている鍵データ k_{i0} , k_{i1} , k_{i2} とともに非線形関数部304 への入力データとなる。ブロックデータ R_i は、鍵データ k_{i0} による第1鍵依存線形変換部341 により例えば排他的論理和がとられてデータ $R_i^* = R_i \oplus k_{i0}$ に線形変換される。次に、この変換されたデータ R_i^* は分割部342 において例えば8ビットづつの4つのデータ in_0 , in_1 , in_2 , in_3 にビット分割される。4つのデータ in_0 , in_1 , in_2 , in_3 は、それぞれ非線形変換部343₀, 343₁, 343₂, 343₃において、データ mid_{00} , mid_{01} , mid_{02} , mid_{03} に非線形変換された後、第2鍵依存線形変換部344 に入力される。

第2鍵依存線形変換部344 では、4つのデータ系統から入力されたデータ mid_{00} , mid_{01} , mid_{02} , mid_{03} を系統間で互いに線形処理（排他的論理和）して新たな4つの系統のデータとし、更にこれらの系列のデータを鍵データ k_{i1} の4つの部分により

それぞれ線形処理（排他的論理和）して4つの系統のデータ $mid_{i0}, mid_{i1}, mid_{i2}, mid_{i3}$ を出力する。これら4つのデータは非線形変換部345₀, 345₁, 345₂, 345₃に入力され、それぞれデータ $out_0, out_1, out_2, out_3$ とされる。これら4つの出力データは結合部346で結合されてデータ Y_i^* とされ、更に第3鍵依存線形変換部347においてデータ Y_i^* と鍵データ k_{i2} との線形処理によりデータ Y_i を生成して出力する。

上述の第2鍵依存線形変換部344は、データ $mid_{00}, mid_{01}, mid_{02}, mid_{03}$ に対応して設けられたデータ処理系統30₀, 30₁, 30₂, 30₃間でこの発明によるアルゴリズムに従ってデータの排他的論理和をとることにより、図4のデータ変換装置のラウンド数を増やすことに安全性を高めるよう構成される。図4のデータ拡散装置の差分解読及び線形解読に対する安全性は、各ラウンドの非線形関数部304の構成に依存し、特に図5に示す非線形関数部301のが図6に示すような基本構成のとき、 m ビットの入力データがそれぞれ与えられる n 個の非線形変換部（S-box）から成る第1非線形変換部343と、それらの n 個の出力を線形変換する線形変換部344Aと、その n 個の m ビット出力をそれぞれ非線形変換する n 個の非線形変換部（S-box）から成る第2非線形変換部345とによって安全性が決まる。特に、いかにして差分解読及び線形解読に対する耐性が強い最適な線形変換部344Aを構成するかが重要である。この発明によれば、線形変換部344Aを{0, 1}上の $n \times n$ 行列 P として表現し、その行列 P の要素を最大差分確率・線形確率 p, q が最小となるように決定していくことにより、最適な線形変換部344Aを構成する。ただし、第2鍵依存線形変換部344に含まれる副鍵 k_{i1} による変換部は行列 P により決定した線形変換部344Aに対し図7に示すように鍵依存変換部344Bとして付加することとする。

なお、”最適である”とは、上記の構造を有する線形変換部344Aの中で差分解読・線形解読に対する耐性が最も強くなるということであり、必ずしも他の指標、例えばアバランシュ性などについても最適であることを意味するわけではない。しかし、経験的にいえば、一般に段数（ラウンド数）を増やすだけでも差分解読・線形解読以外の攻撃を容易に回避できる一方、差分解読・線形解読については注意深くラウンド関数の特性を検討しなければ段数を多少増やしただけで攻撃を回避できるかどうかはわからない。そこで、この発明では、ラウンド関数の差分解

読・線形解読に対する耐性を最も重視し、最適な線形変換部344Aを求める。

この発明によれば、図6において上述のように線形変換部344Aを $\{0, 1\}$ 上の $n \times n$ 行列 P として表現する。この行列 P は、mビット単位での線形変換であり、その線形変換部344Aが排他的論理和XORだけ構成できることを意味する。即ち、この変換を次式

$$z'_i = \bigoplus_{j=0}^{n-1} t_{ij} z_j \quad (1)$$

で表すことができる。特に、 $m = 8$ の場合はバイト単位での線形変換となり、ワード幅が8ビット以上のいずれのプラットフォームにおいても効率的な実装が可能となる。

例えば $n = 4$ の場合の具体例として、次式

$$\begin{bmatrix} z'_0 \\ z'_1 \\ z'_2 \\ z'_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad (2)$$

で表される 4×4 行列 P_e を説明する。行列 P_e を用いて構成したラウンド関数は以下の性質を有する。ただしS-boxは全単射であると仮定する。上記行列式により規定される z'_0, z'_1, z'_2, z'_3 はそれぞれ以下の演算を表している。

$$z'_0 = 0 \cdot z_0 \oplus 1 \cdot z_1 \oplus 1 \cdot z_2 \oplus 1 \cdot z_3 = z_1 \oplus z_2 \oplus z_3 \quad (3-1)$$

$$z'_1 = 1 \cdot z_0 \oplus 0 \cdot z_1 \oplus 1 \cdot z_2 \oplus 1 \cdot z_3 = z_0 \oplus z_2 \oplus z_3 \quad (3-2)$$

$$z'_2 = 1 \cdot z_0 \oplus 1 \cdot z_1 \oplus 1 \cdot z_2 \oplus 0 \cdot z_3 = z_0 \oplus z_1 \oplus z_2 \quad (3-3)$$

$$z'_3 = 1 \cdot z_0 \oplus 1 \cdot z_1 \oplus 1 \cdot z_2 \oplus 1 \cdot z_3 = z_0 \oplus z_1 \oplus z_2 \oplus z_3 \quad (3-4)$$

差分解読及び線形解読に対するラウンド関数の耐性は、active s-boxの最少個数 n_d, n_i から決めることができ、これらの値は行列 P を決定した時点で決まる値である（付録参照）。差分解読法では入力差分値 Δx が非零であるs-boxをactive s-boxと呼び、線形解読法では出力マスク値 Δy が非零であるs-boxをactive s-boxと呼ぶ。

一般に、ある行列 P が与えられたとき、それに対応する線形変換部344Aの構成

は複数存在する。なぜなら、行列 P は線形変換部344Aの入力データと出力データとの関係を表しているだけであり、線形変換部344Aの具体的構成を規定しているわけではない。線形変換部344Aの構成が異なっても、線形変換部344Aの入力データと出力データとの関係を表す行列 P が同じであるならば、それらは同じ特性を持つものと判断できる。従って、以下では差分解読及び線形解読に対する耐性が高く、かつアバランシュ性がよくなるような行列 P を決めた後に線形変換部344Aの構成を決めるにすることにする。その方が、線形変換部344Aの個々の構成について最適な特性を有するかどうかを調べて選択していくより効率よく、最適な特性を有する線形変換部344Aを見つけることができるためである。

$n \times n$ 行列 P の要素は、差分特性に注目して以下の探索アルゴリズムによって決定される。

Step 1 : 安全性閾値 T (T は $2 \leq T \leq n$ なる整数) を設定する。

Step 2 : ハミング重みが $T-1$ 以上となる列ベクトルの集合 C を用意する。具体的には要素"1"の数が $T-1$ 以上の n 次元列ベクトルを n 個以上用意する。

Step 3 : 集合 C から n 個の列ベクトルの組 P_c を選択する。全ての組の検査を完了するまで以下の処理を繰り返す。

Step 3-1 : 列ベクトルの組 P_c について、 n_d を求める。このことを $n_d(P_c)$ と表す。

Step 3-2 : $n_d(P_c) \geq T$ ならば、その n 個の列ベクトルで構成される行列 P_c を候補行列として採用する。

Step 4 : 全ての候補行列の中で、最大の n_d の値を与えた行列 P と $n_d(P)$ の値を出力する。

上記探索アルゴリズムによる候補行列を採用するならば、 n_d が T 以上の値を持つことが保証される。 n_d が最大となるような行列 P を求めるには上記探索アルゴリズムを実行する毎に T を $T=n, n-1, \dots, 3, 2$ の順に 1 ずつ減らしていくことで、効率よく行列 P を発見できる。

上記探索アルゴリズムにおいて、差分解読及び線形解読に対しある程度満足いく耐性が得られるのであれば、Step 3-2まで実行して $n_d(P_c) \geq T$ となる行列を目的

とする行列 P として使用してもよい。あるいは、Step 1を実行し、Step 2でハミング重みが $T-1$ 以上となるベクトルを n 個選択し、それにより構成した行列 P_c を目的とする行列 P として使用してもよい。

線形変換部344Aの入力マスク値は、線形変換部344Aの出力マスク値の排他的論理和によって表現可能であるので、差分特性の場合と同様に、ある行列として表現できる。幾つかの線形変換部344Aの構成について、差分特性に注目した行列と線形表現に注目した行列との関係を調べた結果、以下の 2 つの予測ができた。

予測 1：線形変換部344Aが $\{0, 1\}$ 上の $n \times n$ 行列 P で与えられたとする。このとき、線形変換部344Aの入力差分値 Δz と出力差分値 $\Delta z'$ との関係（差分値パス）は行列 P で与えられ、入力マスク値 Γz と出力マスク値 $\Gamma z'$ との関係（マスク値パス）は転置行列 P^T で与えられる。即ち、

$$\Delta z' = P \Delta z \quad (4)$$

$$\Gamma z = P^T \Gamma z' \quad (5)$$

と表すことができる。

予測 2：行列 P を用いた差分値パスにおけるactive s-boxの最少個数 n_a と行列 P の転置行列 P^T を用いたマスク値パスにおけるactive s-boxの最少個数 n_i は同じである。

上記予想 2 から、前述の探索アルゴリズムによる候補行列を採用するならば、 n_i も T 以上の値を持つ。例えば先の行列 P_E の場合、差分値パスに対する行列 P_E と、マスク値パスに対する行列 P_E^T は以下のようになる。

$$P_E = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \Leftrightarrow P_E^T = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (6)$$

この 2 つの行列について、それぞれ $n_a=3$, $n_i=3$ であることを証明することができる（付録参照）。

行列 P が与えられたとき、それに対応する線形変換部344Aの構成を決めるアルゴリズムを以下に示す。ここでは、

(1) 排他的論理和(XOR) の個数を最少にする、又は
 (2) 同じような副構成が繰り返し現れる、
 を満たすようにする。

Step 1：行列Pにおいて、2つの行を選択し、一方の行（a行）にもう一方の行（b行）を加える（基本演算と呼ぶ）。

Step 2：基本演算のみを用いて、行列Pが単位行列Iになるまで行を選択し、基本演算を行うことを繰り返し、そのときに行った基本演算の回数を数え、基本演算の回数が最少となるような行列変形手順を求める。

Step 3：線形変換部344Aを構成するために、Step 2で求めた変形手順の逆順に、そのときの基本演算で選択したa行、b行に相当するAライン、BラインとのXOR結線を行う。

この様にして決められた線形変換部344Aを有する第2鍵依存形線形変換部344の具体例を例えば図7に示す。線形変換部344Aにおいて、データ mid_{00} , mid_{01} , mid_{02} , mid_{03} はそれぞれ処理系 30_0 ～ 30_3 に入力され、処理系 30_0 で mid_{00} と mid_{01} との排他的論理和がXOR回路 31_0 でとられ、また処理系 30_2 で mid_{02} とXOR回路 31_0 の出力との排他的論理和がXOR回路 31_2 でとられ、更にその回路 31_2 の出力は処理系 30_1 のXOR回路 31_1 で mid_{01} との排他的論理和がとられる。

処理系 30_3 でXOR回路 31_0 の出力と mid_{03} との排他的論理和がXOR回路 31_3 でとられ、処理系 30_1 でXOR回路 31_1 の出力とXOR回路 31_3 の出力との排他的論理和がXOR回路 32_1 でとられ、処理系 30_0 でXOR回路 32_1 の出力とXOR回路 31_0 の出力との排他的論理和がXOR回路 32_0 でとられる。

XOR回路 32_0 , 32_1 , 31_2 , 31_3 の各出力と副鍵データ k_{110} , k_{111} , k_{112} , k_{113} との各排他的論理和が鍵依存変換部344BのXOR回路 35_0 ～ 35_3 でそれぞれとられて、それぞれ mid_{10} , mid_{11} , mid_{12} , mid_{13} を出力する。つまりデータ mid_{00} , mid_{01} , mid_{02} , mid_{03} は相互に関連づけられた後、それぞれ各8ビットの副鍵データ k_{110} , k_{111} , k_{112} , k_{113} に依存した線形変換が行われる。論理式で示すと下記の論理演算がなされる。

$$mid_{10} = mid_{00} \oplus mid_{02} \oplus mid_{03} \oplus k_{110} \quad (7-1)$$

$$mid_{11} = mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus k_{111} \quad (7-2)$$

$$mid_{i2} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus k_{i12} \quad (7-3)$$

$$mid_{i3} = mid_{00} \oplus mid_{01} \oplus mid_{03} \oplus k_{i13} \quad (7-4)$$

なお、副鍵 k_{ii} は4つのデータ k_{i10} , k_{i11} , k_{i12} , k_{i13} で構成されている。

図5に示したように、次に、これらデータ mid_{10} , mid_{11} , mid_{12} , mid_{13} は、それぞれ非線形変換部 345_0 , 345_1 , 345_2 , 345_3 において、データ out_0 , out_1 , out_2 , out_3 に非線形変換された後、結合部346において、一つのデータ Y_i^* に結合される。つまり例えば4つの8ビットのデータが1つの32ビットデータにビット結合される。最後に、データ Y_i^* は、鍵データ k_{i2} による第3鍵依存線形変換部347において、例えばデータ Y_i^* を k_{i2} ビット左ローテーションによりデータ Y_i に線形変換され、非線形関数部304からの出力データ Y_i が生成される。非線形変換部 $343_0 \sim 343_3$, $345_0 \sim 345_3$ のそれぞれは、例えばDES暗号のS-boxのようなものであり、例えばROMにより構成され、それぞれ入力データをアドレスとして入力し、対応するデータを読み出すものである。

ここで、非線形変換部 $343_0 \sim 343_3$ は4つ並列に配置されており、その変換処理は相互に関連していないため、これらは並列実行が可能である。また、非線形変換部 $345_0 \sim 345_3$ についても同様のことがいえる。このため、これら両非線形変換はそれぞれ1ステップ（非線形関数部304としては合計2ステップ）で実行することができる。また、非線形変換部 $343_0 \sim 343_3$, $345_0 \sim 345_3$ の差分確率・線形確率が p とすると、第2鍵依存線形変換部344に図7に示したものを用いた場合、非線形関数部304全体における差分確率・線形確率は p^4 となる。従って、データ変換装置全体では段数（ラウンド数）を $3r$ として確率 $P \leq p^{3r}$ で近似表現でき、例えば $r = 4$ （段数12段）とすると $P \leq p^{32}$ である。これはDES暗号に換算すると48段以上に相当し、差分解読法および線形解読法に対して十分安全なデータ変換装置となる。

なお、鍵データ fk , k_{00} , k_{01} , k_{02} , k_{10} , k_{11} , k_{12} , \dots , $k_{(N-1)0}$, $k_{(N-1)1}$, $k_{(N-1)2}$, ek は図4において鍵生成スケジュール部20の鍵入力部320から入力された主鍵情報 K_{key} から鍵データ生成部321によって変換され、鍵記憶部322に蓄積されたデータである。鍵データ生成部321による鍵データの生成は、図1に示したDES暗号

の拡大鍵生成ルーチン部21と同様に構成してもよいし、あるいは図3に示した宮口らの拡大鍵生成ルーチン部21と同様に構成してもよい。

図4に示した初期鍵依存変換部302、最終鍵依存変換部308、及び図5に示した各非線形関数部304内の鍵依存線形変換部341、344、347は鍵に依存する線形変換部であるため、差分解読法および線形解読法以外の解読法に対しても十分な安全性を兼ね備え、最も安全性を重視した暗号装置である。

なお、この発明はこの例に特定されるだけでなく、例えば高速性を望むのであれば、これら初期鍵依存変換部302、最終鍵依存変換部308、鍵依存線形変換部341、344、347については、そのいずれか1つを削除する、又は鍵に依存しない変換手段に変更することが可能である。この場合、差分解読法および線形解読法に対する安全性をそれほど低下させることなく暗号化処理速度の向上が望める。

第2実施例

図4に示した第1実施例と同様の構成を有するデータ変換装置において、図5に示す非線形関数部304として他の機能構成を用いた別の実施例を説明する。この実施例は、図5における非線形変換部343₀、343₁、343₂、343₃の代わりに、それぞれ例えば8ビットの入力in₀～in₃に対して32ビットの拡大データMID₀₀、MID₀₁、MID₀₂、MID₀₃を非線形変換により出力する等価的に図8A～8Dで示す非線形変換部343₀～343₃を使用し、鍵依存線形変換部344として図9に示すものを使用する。

図5に示したのと同様にデータR_iは、鍵記憶部322に蓄積されている鍵データk₀、k_{i1}、k_{i2}と共に非線形関数部304への入力データとなる。データR_iは、鍵データk_{i0}による第1鍵依存線形変換部341で、例えば排他的論理和によりデータR_i^{*}=R_i⊕k_{i0}に線形変換される。次に、データR_i^{*}は分割部342において4つのデータin₀、in₁、in₂、in₃に分割される。4つのデータin₀、in₁、in₂、in₃は、それぞれ図8A～8Dに示す非線形変換部343₀、343₁、343₂、343₃において、データMID₀₀、MID₀₁、MID₀₂、MID₀₃に非線形変換される。第1実施例では、非線形変換部343₀がmビット入力in₀に対し、mビットのデータmid₀₀を出力したのに対し、ここでは、非線形変換部343₀は、第1実施例の図5における非線形変換部343₀と同じmビットの

データ mid_{00} を上位 m ビットとして出力すると共に、下位 m ビットに固定データ "0 0…0₍₂₎" を出力するよう構成された S-box を有し、その上位 m ビットデータ mid_{00} を布線により 3 系統に出力すると共に m ビットの "00…0₍₂₎" を出力するよう構成されている。即ち、非線形変換部 343₀' は m ビットのデータ in_0 を $4m$ ビットのデータ

$$MID_{00} = [mid_{00}, 00…0_{(2)}, mid_{00}, mid_{00}] \quad (8-1)$$

に変換する手段である。同様に、非線形変換部 343₁', 343₂', 343₃' はそれぞれ in_1 , in_2 , in_3 から

$$MID_{01} = [00…0_{(2)}, mid_{01}, mid_{01}, mid_{01}] \quad (8-2)$$

$$MID_{02} = [mid_{02}, mid_{02}, mid_{02}, 00…0_{(2)}] \quad (8-3)$$

$$MID_{03} = [mid_{03}, mid_{03}, 00…0_{(2)}, mid_{03}] \quad (8-4)$$

に変換する手段である。式(8-1)で表されるデータ MID_{00} の決め方は、予め図 7 の線形変換部 344Aにおいてデータ mid_{00} 以外のデータ mid_{01} , mid_{02} , mid_{03} 全てをそれぞれ "00…0₍₂₎" とおいた場合に線形変換部 344A の出力 4 系統に得られるデータ全体を MID_{00} とすればよい。以下同様にして式(8-1), (8-2), (8-3) で表されるデータ MID_{01} , MID_{02} , MID_{03} についてもそれぞれ容易に決めることができる。これら非線形変換部 343₀' ~ 343₃' はそれぞれデータ in_0 , in_1 , in_2 , in_3 をアドレスとし、データ MID_{00} , MID_{01} , MID_{02} , MID_{03} を直接読み出す変換テーブルとしてメモリにより構成してもよい。

次いで、これらデータ MID_{00} ~ MID_{03} は図 9 に示す鍵データ k_{i1} による第 2 鍵依存線形変換部 344 に入力される。 MID_{00} と MID_{01} は XOR 回路 41 で排他的論理和がとられ、 MID_{02} と MID_{03} は XOR 回路 42 で排他的論理和がとられ、 XOR 回路 41, 42 の両出力の排他的論理和が XOR 回路 43 でとられ、 XOR 回路 43 の出力と鍵データ k_{i1} との排他的論理和が XOR 回路 44 でとられる。 XOR 回路 44 の出力 MID_1 は各 m ビットの出力 mid_{10} , mid_{11} , mid_{12} , mid_{13} にビット分割されて出力される。結局、この第 2 鍵依存線形変換部 344 は次式

$$MID_1 = MID_{00} \oplus MID_{01} \oplus MID_{02} \oplus MID_{03} \oplus k_{i1} \quad (9)$$

の演算で入力を線形変換する。

この線形変換演算による出力 $MID_1 = [mid_{10}, mid_{11}, mid_{12}, mid_{13}]$ の成分はそれぞ

れ次式で表される：

$$mid_{10} = mid_{00} \oplus mid_{02} \oplus mid_{03} \oplus k_{i10} \quad (10-1)$$

$$mid_{11} = mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus k_{i11} \quad (10-2)$$

$$mid_{12} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus k_{i12} \quad (10-3)$$

$$mid_{13} = mid_{00} \oplus mid_{01} \oplus mid_{03} \oplus k_{i13} \quad (10-4)$$

これは、図7における線形変換式(7-1)～(7-4)と等価な線形変換である。このようにして第1実施例と同じデータ mid_{10} , mid_{11} , mid_{12} , mid_{13} が生成される。なお、 k_{ii} は4つのデータ k_{i10} , k_{i11} , k_{i12} , k_{i13} で構成されている。

次いで、データ mid_{10} , mid_{11} , mid_{12} , mid_{13} は、図5に示したのと同様にそれぞれ非線形変換部345₀, 345₁, 345₂, 345₃において、データ out_0 , out_1 , out_2 , out_3 に非線形変換された後、結合部346において、4つのデータ out_0 , out_1 , out_2 , out_3 が1つのデータ Y_i^* に結合される。最後に、データ Y_i^* は、鍵データ k_{i2} による第3鍵依存線形変換部347において、例えば k_{i2} ビット左ローテーションによりデータ Y_i に線形変換され、非線形関数部304からの出力データ Y_i が生成される。

図8A～8D, 9に示した第2実施例において、図8A～8Dの非線形変換部343₀～343₃は第1実施例と同様にそれぞれ各8ビットのデータ mid_{00} ～ mid_{03} を出力するS-boxのみで構成し、図8A～8Dで示した布線と8ビットの“00…0”を出力するレジスタを図9の鍵依存線形変換部344内に設け、その中でデータ MID_{00} ～ MID_{03} を生成するようにもよい。

この第2実施例における第2鍵依存線形変換部344では、図9に示したように4つ排他的論理和により図7と等価な線形変換（図7では排他的論理和の個数は10個）を実現しているため、第1実施例より高速な変換が可能になる。

また、第1実施例と同様に、非線形変換部343₀～343₃と345₀～345₃は4つずつ並列に配置されており、その非線形変換処理は相互に関連していないため、これらはすべて並列実行が可能である。更に、非線形変換部343₀～343₃, 345₀～345₃の差分確率・線形確率がpとすると、非線形関数部304全体における差分確率・線形確率は p^4 となる。

第3実施例

第1実施例と同様に、図4に示す機能構成を有するデータ変換装置において、その非線形関数部304の内部の機能構成として更に別のものを用いた実施例を説明する。

図5に示したように例えば32ビットのデータ R_i は、鍵記憶部322に蓄積されている鍵データ k_{i0} , k_{i1} , k_{i2} とともに非線形関数部304へ入力される。データ R_i は、鍵データ k_{i0} による第1鍵依存線形変換部341で、例えば排他的論理和によりデータ $R_i^* = R_i \oplus k_{i0}$ に線形変換される。次に、データ R_i^* は分割部342において例えば8ビットの4つのデータ in_0 , in_1 , in_2 , in_3 に分割される。

非線形変換部343₀において、例えば図1-0に示すように、 in_0 は更に例えば4ビットの2つのサブブロック in_{00} と in_{01} に分割された後、 in_{00} はサブ非線形変換部51によりデータ mid_{000} に変換され、また in_{01} との排他的論理和がXOR回路52でとられ、その出力 $in_{00} \oplus in_{01}$ はサブ非線形変換部53によりデータ mid_{001} に変換される。ついで、これら出力 mid_{000} と mid_{001} の排他的論理和がXOR回路54でとられ、そのXOR回路54の出力と mid_{001} を統合して mid_{00} に変換する。つまり、非線形変換部343₀は入力 in_0 を2つのサブブロックに分割し、これら2つのサブブロックについて、それぞれ線形変換と非線形変換を行い、その2つの出力サブブロックを統合して非線形変換部の出力とする。同様に、 in_1 , in_2 , in_3 についても、それぞれ2つの非線形変換部と2つの排他的論理和回路とよりなる図1-0に示した機能構成の非線形変換部343₁, 343₂, 343₃を用いて、データ mid_{01} , mid_{02} , mid_{03} に変換する。

変換されたデータ mid_{00} , mid_{01} , mid_{02} , mid_{03} は、図7に示した鍵データ k_{i1} による第2鍵依存線形変換部344に入力される。この変換部344によりそれぞれ前述の式(7-1)～(7-4)の演算が行われる。

次いで、データ mid_{10} は、非線形変換部345₀においても、例えば図1-0に示したのと同様の機能構成により、更に2つのサブブロック mid_{100} と mid_{101} に分割され、 mid_{100} はサブ非線形変換部51によりデータ out_{00} に変換され、一方、データ mid_{100} と mid_{101} はXOR回路52により排他的論理和がとられ、その出力 $mid_{100} \oplus mid_{101}$ は非線形変換部53によりデータ out_{01} に変換される。ついで、データ out_{00} と out_{01} の排他的論理和がXOR回路54によりとられ、その出力 $out_{00} \oplus out_{01}$ と out_{01} を統合して out_0 論理和がXOR回路54によりとられ、その出力 $out_{00} \oplus out_{01}$ と out_0 を統合して out_0

に変換する。同様に、 mid_{11} , mid_{12} , mid_{13} についても、それぞれ2つのサブ非線形変換部51, 53と2つの排他的論理和回路52, 54によりなる図10に示したのと同様の機能構成の非線形変換部345₁, 345₂, 345₃によりデータout₁, out₂, out₃に変換される。

これら非線形変換された4つのデータout₀, out₁, out₂, out₃は、結合部346において一つのデータ Y_i^* に結合される。最後に、データ Y_i^* は、鍵データ k_{i2} による第3鍵依存線形変換部347において、例えば k_{i2} ビット左ローテーションによりデータ Y_i に線形変換され、非線形関数部304からの出力データ Y_i が生成される。

この様に、この第3実施例においては、非線形変換部343₀～343₃, 345₀～345₃では、それぞれその内部で入力データが2つのデータに分割されて2つのサブ非線形変換部（図10では51, 53）により非線形変換される。このため、これら16個の各非線形変換部がそれぞれ扱うことのできる2倍のビット長を、非線形変換部343₀～343₃, 345₀～345₃への入力データとすることができます。例えば、サブ非線形変換部51, 53を8ビット長のS-boxとすると、非線形変換部343₀～343₃, 345₀～345₃への各入力データは16ビット長となり、非線形変換関数304への入力データは64ビット長となる。これにより、図4の構成を有するデータ変換装置として、ブロック長を128ビットとすることができます。

また、サブ非線形変換部51, 53対応のものは8つずつ並列に配置されており、その非線形変換処理は相互に関連していないため、これらはすべて並列実行が可能である。更に、サブ非線形変換部51, 53の差分確率・線形確率がpとすると、非線形関数部304全体における差分確率・線形確率は p^4 となる。

上述において第1鍵依存線形変換部341, 第2鍵依存線形変換部344及び第3鍵依存線形変換部347は必ずしも鍵依存とすることなく、つまり鍵データを入力することなく、サブデータ内での線形変換を行ってもよい。

前述の各実施例ではハードウェア構成として説明したが、そのデータ処理をプログラムに従って実行するソフトウェアにより実現してもよい。例えば、その処理手順の要部をフローチャートとして図11に示す。図11は図4の処理全体に対応する処理手順を表している。

ステップS 1：処理繰り返し回数を表す変数 i を 0 に初期化する。

ステップS 2：入力平文を初期変換してから左右ブロックデータ L_i, R_i に分割する。

ステップS 3：副鍵 k_i を使って非線形関数により右ブロックデータ R_i を処理してブロックデータ Y_i を生成する。

ステップS 4：左ブロックデータ L_i をブロックデータ Y_i で線形処理してブロックデータ L'_i を生成する。

ステップS 5：右ブロックデータ R_i を新しい左ブロックデータ L_i とし、ブロックデータ L'_i を新しい右ブロックデータ R_i とする。

ステップS 6： i を 1 だけインクリメントする。

ステップS 7： i が N に達したか判定し、達してなければステップS 3 に戻り、前述と同様の処理ステップS 3～S 7 を繰り返す。

ステップS 8：前ステップS 7 で i が N に達していると判定されると、左右データ L_i, R_i を結合し、更に最終変換した結果を拡散データ C として出力する。

図1 1 のステップS 3 の処理の詳細は図5の非線形関数部304 による処理に対応し、その処理手順を図1 2 に示す。

ステップS 3 1：右データ R_i に対し第1の鍵依存線形変換を行い、データ R'_i とする。

ステップS 3 2：データ R'_i を m ビットずつ n 分割してデータ $in_0, in_1, \dots, in_{n-1}$ を生成する（例えば $m=8, n=4$ である）。

ステップS 3 3：データ $in_0, in_1, \dots, in_{n-1}$ をそれぞれアドレスとして n 個の第1のS-box からそれぞれデータ $mid_{00}, mid_{01}, \dots, mid_{0(n-1)}$ を読み出す。

ステップS 3 4：データ $mid_{00} \sim mid_{0(n-1)}$ に対し副鍵 k_{i1} を使った鍵依存線形変換を行い、データ $mid_{10} \sim mid_{1(n-1)}$ を生成する。

ステップS 3 5：データ $mid_{10} \sim mid_{1(n-1)}$ をそれぞれアドレスとして n 個の第2のS-box からデータ $out_0 \sim out_{n-1}$ を読み出す。

ステップS 3 6：データ $out_0 \sim out_{n-1}$ を結合してデータ Y'_i を得る。

ステップS 3 7：データ Y'_i に対し、第3の鍵依存線形変換を行ってデータ Y_i を

生成し、出力する。

上記ステップS3～S4の演算は、式(7-1)～(7-4)による演算でもよいし、又は式(8-1)～(8-4)の定義を使った式(9)による演算でもよい。更に、図1～1ではステップS3乃至S7をラウンド数だけ繰り返し実行する処理手順の場合を示したが、図3に示したN段のラウンド処理部38₀～38_{N-1}のそれぞれの処理をそのままプログラミングしてこの発明のデータ拡散を実現してもよい。

第4実施例

前述の図4に示した第1実施例は、その非線形関数部304(図5)の第2鍵依存線形変換部344を構成する図6で説明した基本線形変換部344Aを4×4の行列で表す場合(即ち、4入力、4出力の場合)の実施例であったが、以下に示す第4実施例では、線形変換部344Aを8×8行列で表す場合について説明する。

図13は、この発明の第4実施例を示すデータ変換装置における、暗号化処理手順の機能構成を示したものである。この構成自体は図4に示した第1実施例のものと同じであるが、異なる点は、データのビット数と、非線形関数部304におけるデータの分割数nである。

入力データMは、鍵記憶部322に蓄積されている鍵データfkによる初期鍵依存変換部302で変換された後、初期分割部303で左、右ブロックデータL₀、R₀に分割される。例えば128ビットのデータが64ビットずつのブロックデータL₀、R₀に分割される。初期鍵依存変換部302では、例えば鍵データfkと入力データMとの排他的論理和や鍵データfkによる入力データMのビットローテーション(ビット回転)などの線形変換、もしくは乗算などを組み合わせた非線形変換などが行われる。

右ブロックデータR₀は、鍵記憶部322に蓄積されている鍵データk₀₀、k₀₁、k₀₂とともに非線形関数部304に入力され、非線形関数部304で非線形変換処理が行われてデータY₀に変換される。データY₀とデータL₀は線形演算部305で線形演算され、データL₀^{*}に変換される。データL₀^{*}とデータR₀は交換部306でデータ位置の交換が行われ、L₁←R₀、R₁←L₀^{*}とされ、L₁、R₁が次の第1段ラウンド処理部381に入力

される。

以下、第 i 段ラウンド処理部38_i ($i=0, 1, \dots, N-1$)において、左右ブロックデータ L_i, R_i について上記と同様の処理を繰り返し行う。即ち、左右ブロックデータ L_i, R_i について、右ブロックデータ R_i は、鍵記憶部322 に蓄積されている鍵データ k_{i0}, k_{i1}, k_{i2} と共に非線形関数部304 に入力され、非線形関数部304 で非線形変換処理が行われて、ブロックデータ Y_i に変換される。ブロックデータ Y_i と L_i は線形演算部305 で演算されてデータ L_i^* に変換される。データ L_i^* とデータ R_i は交換部306 でデータ位置の交換が行われ、 $L_{i+1} \leftarrow R_i, R_{i+1} \leftarrow L_i^*$ のように交換される。線形演算部305 は例えば排他的論理和演算を行うものである。

データ変換装置としての安全性を確保するための適切なラウンド繰り返し回数を N とすると、繰り返し処理の結果、左右ブロックデータ L_N, R_N が得られる。これらのブロックデータ L_N, R_N を最終結合部307 で結合し、つまり例えば各 64 ビットのブロックデータ L_N, R_N を結合して 128 ビットのデータとし、その後、鍵記憶部322 に蓄積されている鍵データ ek による最終鍵依存変換部308 で変換し、出力部309 から暗号文として拡散データ C を出力する。

復号については、暗号化処理手順と逆の手順をたどることによって、暗号文 C から平文 M が得られる。特に、最終鍵依存変換部308 が初期鍵依存変換部302 の逆変換になっているならば、図 13において入力データの代りに暗号文データを入力し、鍵データを図 13 とは逆に、 $ek, k_{(N-1)0}, k_{(N-1)1}, k_{(N-1)2}, \dots, k_{10}, k_{11}, k_{12}, k_{00}, k_{01}, k_{02}, fk$ を順次与えればよい。

次に、非線形関数部304 の内部を詳細に説明する。図 14 は、非線形関数部304 の内部の機能構成を抜き出して示したものである。

右ブロックデータ R_i は、鍵記憶部322 に蓄積されている鍵データ k_{i0}, k_{i1}, k_{i2} と共に非線形関数部304 への入力データとなる。右ブロックデータ R_i は、副鍵データ k_{i0} による第 1 鍵依存線形変換部341、例えば排他的論理和によりデータ $R_i^* = R_i \oplus k_{i0}$ に線形変換される。この変換されたデータ R_i^* は分割部342 において例えば $n = 8$ 個のデータ $in_0, in_1, in_2, \dots, in_7$ に分割される。8 つのデータ $in_0 \sim in_7$ は、それぞれ非線形変換部343₀ ∼343₇ において、データ $mid_{00} \sim mid_{07}$ に非線形変換さ

れた後、鍵データ k_{i1} による第 2 鍵依存線形変換部344 に入力される。

第 2 鍵依存線形変換部344 では、8 つのデータ系統から入力されたデータ mid_{i00} , mid_{i01} , mid_{i02} , \dots , mid_{i07} を系統間で互いに線形処理（排他的論理和）して新たな 8 つの系統のデータとし、更にそれらの系列のデータを鍵データ k_{i1} の 8 つの部分によりそれぞれ線形処理（排他的論理和）して 8 つの系統のデータ mid_{i10} , mid_{i11} , mid_{i12} , \dots , mid_{i17} を出力する。これら 8 つのデータは非線形変換部345₀, 345₁, 345₂, \dots , 345₇ に入力され、それぞれデータ out_0 , out_1 , out_2 , \dots , out_7 とされる。これら 8 つの出力データは結合部346 で結合されてデータ Y_i^* とされ、更に第 3 鍵依存線形変換部347 においてデータ Y_i^* と鍵データ k_{i2} との線形処理によりデータ Y_i を生成して出力する。

第 2 鍵依存線形変換部344 は、図 6 で説明したと同様に $n \times n$ 行列で表される線形変換部344A を含んでおり、この第 4 実施例では $n = 8$ である。ただし、線形変換部は全単射であると仮定する。即ち、 $\text{rank}(P) = 8$ である。以下に、第 1 実施例で説明したと同様に、 n_d が最大となるような 8×8 行列 P を求める。ここでは、安全閾値 T を $T=8, 7, \dots$ と順次減らし、各値毎に以下のアルゴリズムを実行する。

Step 1 : 安全閾値 T (T は $2 \leq T \leq n$ なる整数) を設定する。

Step 2 : ハミング重みが $T-1$ 以上となる列ベクトルの集合 C を用意する。

Step 3 : 集合 C から 8 個の列ベクトルの組 P_c を選択する。このとき、 $\text{rank}(P_c) \neq 8$ ならば、その P_c は候補としない。

Step 3-1 : P_c に対する n_d を以下のようにして求める。

・任意の 2 列 (a, b 列) について

$$n_{d0} = 2 + \min_{(a, b)} \# \{(t_{ia}, t_{ib}) \mid t_{ia} \oplus t_{ib} \neq 0, 0 \leq i < 8\}$$

・任意の 3 列 (a, b, c 列) について

$$n_{d1} = 3 + \min_{(a, b, c)} \# \{(t_{ia}, t_{ib}, t_{ic}) \mid t_{ia} \oplus t_{ib} \oplus t_{ic} \neq 0, 0 \leq i < 8\}$$

$$n_{d2} = 3 + \min_{(a, b, c)} \# \{(t_{ia}, t_{ib}, t_{ic}) \mid (0, 0, 0), (1, 1, 1) を除く, 0 \leq i < 8\}$$

・任意の4列 (a, b, c, d列) について

$$n_{d3} = 4 + \min_{(a, b, c, d)} \# \{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid \begin{array}{l} (0,0,0,1), (0,0,1,0), (0,1,0,0), (1,0,0,0) \\ (0,1,1,1), (1,0,1,1), (1,1,0,1), (1,1,1,0) \end{array}, 0 \leq i < 8\}$$

$$n_{d4} = 4 + \min_{(a, b, c, d)} \# \{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0), (1,1,0,0), (0,1,1,1), (1,0,1,1) を除く, 0 \leq i < 8\}$$

$$n_{d5} = 4 + \min_{(a, b, c, d)} \# \{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0), (1,0,1,0), (0,1,1,1), (1,1,0,1) を除く, 0 \leq i < 8\}$$

$$n_{d6} = 4 + \min_{(a, b, c, d)} \# \{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0), (1,0,0,1), (0,1,1,1), (1,1,1,0) を除く, 0 \leq i < 8\}$$

$$n_{d7} = 4 + \min_{(a, b, c, d)} \# \{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0), (0,1,1,0), (1,0,1,1), (1,1,0,1) を除く, 0 \leq i < 8\}$$

$$n_{d8} = 4 + \min_{(a, b, c, d)} \# \{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0), (0,1,0,1), (1,0,1,1), (1,1,1,0) を除く, 0 \leq i < 8\}$$

$$n_{d9} = 4 + \min_{(a, b, c, d)} \# \{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0), (0,0,1,1), (1,1,0,1), (1,1,1,0) を除く, 0 \leq i < 8\}$$

$$\cdot n_d = \min \{n_{di} \mid 0 \leq i \leq 9\}$$

直感的にいえば、式 $n_{d0} \sim n_{d9}$ は、第1非線形変換部343のactive s-boxの個数(右辺第1項)が決められたとき、第2非線形変換部345で最小いくつ以上(右辺第2項)のs-boxがactiveになり、そのときのactive s-boxの合計がいくつ以上(左辺)になるのかを表している。例えば、第1非線形変換部343で2つのactive s-boxがある場合、その差分値をそれぞれ Δz_a , Δz_b と表すことができる。このとき、

$$[\Delta z'_{-i}] = [t_{ia} \Delta z_a \oplus t_{ib} \Delta z_b] \quad (0 \leq i < 8) \quad (11)$$

となる。特に、 $\Delta z_a = \Delta z_b$ とすると、

$$[\Delta z'_{-i}] = [(t_{ia} \oplus t_{ib}) \Delta z_a] \quad (0 \leq i < 8) \quad (12)$$

となる。従って、この場合のactive s-boxの最少個数は n_{d0} で与えられることになる。

上述の探索アルゴリズムにより、行列Pを探索した結果、 $n_d \geq 6 = T$ となるような行列は存在せず、また、 $n_d = 5 = T$ となる行列は10080個存在した。その結果、これら

の行列 P を用いたラウンド関数の差分解読法に対する耐性は $p \leq p_s^5$ となる。また、線形解読法に対する耐性も $q \leq q_s^5$ となる。

上記の10080 個の行列 P について、その構成を決める。全数探索的に構成を決めるることは例えば16 個のXOR結線を使うとして、 $(8 \times 7)^{16} \approx 2^{93}$ 程度の計算量が必要であり、実行不可能である。そこで、図15Aに示すように、線形変換部344Aの内部が8入力4出力のボックス $B_1 \sim B_4$ で構成される構造に限定する。各ボックスの内部構成は図15Bに示すように、4つのXOR回路で構成されており、全ての入力ラインが1回ずつXOR回路を通過するのもとする。従って、線形変換部344A全体では、16個のXOR回路で構成されることになる。このとき、 $(4 \times 3 \times 2 \times 1)^4 \approx 2^{16}$ 程度の計算量となり、十分に全数探索が実行可能となる。

図15Aでは左右各4系統のラインに対し4つの変換ボックス $B_1 \sim B_4$ が交互に挿入されているが、これらのラインは任意に選択した4ラインと、その残りの4ラインに決めてよい。各変換ボックスにはそれが挿入されている4ラインからの入力と、残りの4ラインからの入力が与えられ、変換結果を前者の4ラインに出力する。

上述の探索アルゴリズムにより得た10080 個の行列のうち、図15の構造を満たしつつ、16回の基本演算(XOR)で単位行列 I となるものがあるかどうかを探した結果、57個の構成が見つかった。その中の1つの行列 P を次に示す。

$$P = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

この行列を使用した線形変換部344Aの構成例を非線形変換部343, 345と共に図16に示す。図に示すように、第1非線形変換部343を構成する8個のS-boxからの8ラインの左右4系統のラインに対し、4つの変換ボックス $B_1 \sim B_4$ が交互に挿入されており、その結果、各ラインには2個のXOR回路が挿入されている。

第1実施例において 4×4 行列の場合に与えたと同様の予測、即ち図16で構成された線形変換部344Aにおいて、マスク値パスに対する行列が行列Pの転置行列となっていること、及び $n_i=5$ となることが正しく成立するかを以下のように確認することができる。図16で構成された線形変換部344Aに、付録に定理2で示すconcatenation rulesを用いて線形パスを構成することによって、マスク値パスに対する行列 TP は以下のように求められる。

$${}^TP = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (14)$$

このことから、行列 TP が行列Pの転置行列となっていることがわかる。更に、active s-boxの最少個数を調べると、 $n_i=5$ となることが確認できる。

図17は、上述のようにして決定された線形変換部344Aと、更に鍵変換部344Bとを含む第2鍵依存線形変換部344の具体例を示す。

鍵変換部344Bは鍵データ $k_{i10}, k_{i11}, k_{i12}, \dots, k_{i17}$ と線形変換部の出力との排他的論理和演算をXOR回路 $63_0, 63_1, 63_2, \dots, 63_7$ で行い、出力データ $mid_{i0}, mid_{i1}, mid_{i2}, \dots, mid_{i7}$ を出力するように構成されている。例えば、図17に示すような機能的構成により、次の演算がなされる。

$$mid_{i0} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus k_{i10} \quad (15-1)$$

$$mid_{i1} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus k_{i11} \quad (15-2)$$

$$mid_{i2} = mid_{00} \oplus mid_{01} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{07} \oplus k_{i12} \quad (15-3)$$

$$mid_{i3} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{04} \oplus mid_{05} \oplus mid_{07} \oplus k_{i13} \quad (15-4)$$

$$mid_{i4} = mid_{00} \oplus mid_{01} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus k_{i14} \quad (15-5)$$

$$mid_{i5} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{05} \oplus mid_{06} \oplus k_{i15} \quad (15-6)$$

$$mid_{i6} = mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{06} \oplus mid_{07} \oplus k_{i16} \quad (15-7)$$

$$mid_{i7} = mid_{00} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{07} \oplus k_{i17} \quad (15-8)$$

この演算により、データ $mid_{10}, mid_{11}, mid_{12}, \dots, mid_{17}$ を生成する。なお、副鍵 k_{i1} は 8 つのデータ $k_{i10}, k_{i11}, k_{i12}, \dots, k_{i17}$ で構成されている。図 1 7において、 $mid_{00} \sim mid_{07}$ はそれぞれ系路 60₀～60₇ に入力される。

系路 60₄, 60₅, 60₆, 60₇ 上の XOR 回路 61₁, 61₅, 61₆, 61₇ でそれぞれ mid_{01} と mid_{00} 、 mid_{05} と mid_{04} 、 mid_{06} と mid_{02} 、 mid_{07} と mid_{03} との各排他的論理和算がなされる。

系路 60₀, 60₁, 60₂, 60₃ 上の XOR 回路 61₀, 61₁, 61₂, 61₃ でそれぞれ mid_{00} と XOR 回路 61₆ の出力、 mid_{01} と XOR 回路 61₇ の出力、 mid_{02} と XOR 回路 61₄ の出力、 mid_{03} と XOR 回路 61₅ の出力の各排他的論理和算がなされる。

系路 60₄, 60₅, 60₆, 60₇ 上の回路 62₁, 62₅, 62₆, 62₇ で、それぞれ XOR 回路 61₃ と 61₄ の各出力、XOR 回路 61₀ と 61₅ の各出力、XOR 回路 61₁ と 61₆ の各出力、XOR 回路 61₂ と 61₇ の各出力の排他的論理和算がなされる。

系路 60₀, 60₁, 60₂, 60₃ 上の XOR 回路 62₀, 62₁, 62₂, 62₃ で、それぞれ XOR 回路 61₀ と 62₄ の各出力、XOR 回路 61₁ と 62₅ の各出力、XOR 回路 61₂ と 62₆ の各出力、XOR 回路 61₃ と 62₇ の各出力の排他的論理和算がなされる。

更に、系路 60₀～60₇ 上の XOR 回路 63₀～63₇ で、XOR 回路 62₀～62₇ の各出力と鍵データ $k_{i10} \sim k_{i17}$ との排他的論理和算がそれぞれとられ、系路 60₀～60₇ からそれぞれ出力 $mid_{10} \sim mid_{17}$ が output される。つまり出力 $mid_{10} \sim mid_{13}$ は、入力 $mid_{00} \sim mid_{07}$ から選んだ 6 つと鍵データとの排他的論理和算がなされ、出力 $mid_{14} \sim mid_{17}$ は入力 $mid_{00} \sim mid_{07}$ から選んだ 5 つと鍵データとの排他的論理和算がなされる。

次いで、データ $mid_{10}, mid_{11}, mid_{12}, \dots, mid_{17}$ は、図 1 4 でそれぞれ非線形変換部 345₀, 345₁, 345₂, …, 345₇ において、データ $out_0, out_1, out_2, \dots, out_7$ に非線形変換された後、結合部 346 において、8 つのデータ $out_0, out_1, out_2, \dots, out_7$ は 1 つのデータ Y_i^* に結合される。最後に、データ Y_i^* は、鍵データ k_{i2} による第 3 鍵依存線形変換部 347 において、例えば k_{i2} ビット左ローテーションによりデータ Y_i に線形変換され、非線形関数部 304 からの出力データ Y_i が生成される。

非線形変換部 343₀～343₇, 345₀～345₇ のそれぞれは、例えば DES 暗号の S-box のようなものであり、例えば ROM により構成され、それぞれ入力データをアドレスとして入力し、対応するデータを読み出すものである。

ここで、非線形変換部343₀～343₇は8つ並列に配置されており、その非線形変換処理は相互に関連していないため、これらはすべて並列実行が可能である。また、非線形変換部345₀～345₇についても同様のことがいえる。このため、非線形変換はそれぞれ1ステップ（合計2ステップ）で実行することができる。

また、非線形変換部343₀～343₇の差分確率・線形確率がpとすると、第2鍵依存線形変換部344に図17に示した線形変換を用いた場合、非線形関数部304全体における差分確率・線形確率はp³となる。従って、データ変換装置全体では段数を3^rとして確率P≤p^{10r}で近似表現でき、例えばr=4（段数12段）とするとP≤p⁴⁰である。これはDES暗号に換算すると60段以上に相当し、差分解読法および線形解読法に対して十分安全なデータ変換装置となる。なお、第2の鍵依存線形変換部344として図17に示したもの以外の線形変換手段を用いることもできる。その例を図18に示す。この場合は下記の演算がなされる。

$$mid_{10} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i10} \quad (16-1)$$

$$mid_{11} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i11} \quad (16-2)$$

$$mid_{12} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i12} \quad (16-3)$$

$$mid_{13} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i13} \quad (16-4)$$

$$mid_{14} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i14} \quad (16-5)$$

$$mid_{15} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i15} \quad (16-6)$$

$$mid_{16} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i16} \quad (16-7)$$

$$mid_{17} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i17} \quad (16-8)$$

あるいは、図19に示す回路構成としてもよく、この場合は下記の演算がなされる。

$$mid_{10} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i10} \quad (17-1)$$

$$mid_{11} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i11} \quad (17-2)$$

$$mid_{12} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i12} \quad (17-3)$$

$$mid_{13} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i13} \quad (17-4)$$

$$mid_{14} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i14} \quad (17-5)$$

$$mid_{15} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i15} \quad (17-6)$$

$$mid_{16} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus k_{116} \quad (17-7)$$

$$mid_{17} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus k_{117} \quad (17-8)$$

第2鍵依存線形変換部344は図17～19の演算から明らかのように、8つの入力データ $mid_{00}, mid_{01}, mid_{02}, \dots, mid_{07}$ のうち、それぞれ6つの入力データから生成した4つの出力データと、それぞれ5つの入力データから生成した4つの出力データの合計8つの出力データ $mid_{10}, mid_{11}, mid_{12}, \dots, mid_{17}$ を生成するような鍵依存線形変換であり、かつ8つの各入力データ $mid_{00}, mid_{01}, mid_{02}, \dots, mid_{07}$ が少なくとも4つ以上の他の系統の出力データに影響を与える（例えば図17の例であれば、入力データ mid_{00} は6つの出力データ $mid_{11}, mid_{12}, mid_{13}, mid_{14}, mid_{15}, mid_{17}$ に影響を与える）ような線形変換であれば、図17の例について述べたと同様に、非線形関数部304全体における差分確率・線形確率は p^5 となる。

鍵データ $\{fk, k_{00}, k_{01}, k_{02}, k_{10}, k_{11}, k_{12}, \dots, k_{(N-1)0}, k_{(N-1)1}, k_{(N-1)2}, ek\}$ は、主鍵情報 $k e y$ が入力部320より鍵データ生成部321に入力され、この鍵データ生成部321によって変換され、鍵記憶部322に蓄積されたデータである。

鍵データ生成部321による鍵データの生成は図1に示したDES暗号の拡大鍵生成ルーチン部21と同様に構成してもよいし、あるいは米国特許No.4,850.019に示されている拡大鍵生成部と同様に構成してもよい。

また、初期鍵依存変換部302、最終鍵依存変換部308、鍵依存線形変換部341、344、347は鍵に依存する線形変換手段であるため、差分解読法および線形解読法以外の解読法に対しても十分な安全性を兼ね備えたデータ変換装置である。

なお、第4実施例はこの例に特定されるだけでなく、例えば高速性を望むのであれば、これら初期鍵依存変換部302、最終鍵依存変換部308、鍵依存線形変換部341、344、347について、そのいずれかを削除する、または鍵に依存しない変換手段に変更することが可能である。この場合、差分解読法および線形解読法に対する安全性をそれほど低下させることなく、暗号化処理速度の向上が望める。

第5実施例

図13に示した第4実施例と同様の構成を有するデータ変換装置において、非線形関数部304の機能構成の変形例を用いた実施例を説明する。第5実施例の基

$$MID_{04} = [mid_{01}, 00\cdots0_{(2)}, mid_{04}, mid_{01}, mid_{04}, 00\cdots0_{(2)}, 00\cdots0_{(2)}, mid_{04}] \quad (18-5)$$

$$MID_{05} = [mid_{05}, mid_{05}, 00\cdots0_{(2)}, mid_{05}, mid_{05}, mid_{05}, 00\cdots0_{(2)}, 00\cdots0_{(2)}] \quad (18-6)$$

$$MID_{06} = [mid_{06}, mid_{06}, mid_{06}, 00\cdots0_{(2)}, 00\cdots0_{(2)}, mid_{06}, mid_{06}, 00\cdots0_{(2)}] \quad (18-7)$$

$$MID_{07} = [00\cdots0_{(2)}, mid_{07}, mid_{07}, mid_{07}, 00\cdots0_{(2)}, 00\cdots0_{(2)}, mid_{07}, mid_{07}] \quad (18-8)$$

これらのデータ $MID_{00} \sim MID_{07}$ は、第 2 実施例において式(8-1)～(8-4)について説明したと同様な手法により予め決めることができる。即ち、データ MID_{00} は、図 1 7 に示す線形変換部 344Aにおいて、データ mid_{00} 以外の全てのデータ $mid_{01} \sim mid_{07}$ をそれぞれ“00…0₍₂₎”と設定した場合に、線形変換部 344A の 8 系統 $60_0 \sim 60_7$ の出力に得られるデータのセットである。データ MID_{01} は、データ mid_{01} 以外の全てのデータ $mid_{00}, mid_{02} \sim mid_{07}$ をそれぞれ“00…0₍₂₎”と設定した場合に、線形変換部 344A の 8 系統 $60_0 \sim 60_7$ の出力に得られるデータのセットである。以下、データ $MID_{02} \sim MID_{07}$ についても同様である。これら非線形変換部 343₀～343₇ はそれぞれデータ $in_0 \sim in_7$ をアドレスとし、データ $MID_{00} \sim MID_{07}$ を直接読み出す変換テーブルとしてメモリにより構成してもよい。

次いで、図 2 1 に示すように、データ $MID_{00} \sim MID_{07}$ は鍵データ k_{i1} による第 2 鍵依存線形変換部 344 に入力される。第 2 鍵依存線形変換部 344 は、入力データを 2 つずつ互いに排他的論理和をとる XOR 回路 41₁～41₄ と、それらの出力を 2 つずつ互いに排他的論理和をとる XOR 回路 42₁～42₂ と、それらの排他的論理和をとる XOR 回路 43 と、その出力と鍵データ k_{i1} との排他的論理和をとる XOR 回路 44 とから構成されている。これにより、

$$MID_1 = MID_{00} \oplus MID_{01} \oplus MID_{02} \oplus MID_{03} \oplus MID_{04} \oplus MID_{05} \oplus MID_{06} \oplus MID_{07} \oplus k_{i1} \quad (19)$$

が演算される。この出力 MID_1 は 8 個のブロックに分割され、データ $mid_{10}, mid_{11}, mid_{12}, \dots, mid_{17}$ として出力される。結局、第 2 鍵依存線形変換部 344 による線形変換は、m ビットのサブブロック単位として表すと次のようになる：

$$mid_{10} = mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus mid_{06} \oplus k_{i10} \quad (20-1)$$

$$mid_{11} = mid_{00} \oplus mid_{02} \oplus mid_{03} \oplus mid_{05} \oplus mid_{06} \oplus mid_{07} \oplus k_{i11} \quad (20-2)$$

$$mid_{12} = mid_{00} \oplus mid_{01} \oplus mid_{03} \oplus mid_{04} \oplus mid_{06} \oplus mid_{07} \oplus k_{i12} \quad (20-3)$$

$$mid_{13} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{04} \oplus mid_{05} \oplus mid_{07} \oplus k_{i13} \quad (20-4)$$

本構成は、図13に示した第4実施例と同様である。異なる点は、その非線形関数部301を示す図14において、非線形変換部343₀～343₇の構成を、図8A～8Dに示した第2実施例における非線形変換部343₀', 343₁', 343₂', 343₃'等と同様に変形し、拡大データを出力する構成とした点である。また、第2鍵依存線形変換部344の構成を、図9に示したものと同様の構成としている。

図13に示したように、右ブロックデータR_iは、鍵記憶部322に蓄積されている鍵データk_{i0}, k_{i1}, k_{i2}と共に非線形関数部304への入力データとなる。データR_iは、第1鍵依存線形変換部341において、図14で示したのと同様に鍵データk_{i0}と例えば排他的論理和がとられ、データR_i'=R_i⊕k_{i0}に線形変換される。次に、データR_i'は分割部342において8つのデータin₀₀, in₀₁, in₀₂, ..., in₀₇に分割される。8つのデータin₀₀, in₀₁, in₀₂, ..., in₀₇は、それぞれ非線形変換部343₀', 343₁', 343₂', ..., 343₇'において、データMID₀₀, MID₀₁, MID₀₂, ..., MID₀₇に非線形変換される。非線形変換部343₀'は、mビットのデータin₀₀を8×mビットのデータ

$$MID_{00} = [00\cdots 0_{(2)}, mid_{00}, mid_{00}, mid_{00}, mid_{00}, mid_{00}, 00\cdots 0_{(2)}, mid_{00}] \quad (18-1)$$

に変換するよう構成されている。即ち、非線形変換部343₀'は、例えば図20Aに示すように、上位mビットに第4実施例の図14における非線形変換部343₀と同じmid₀₀を出力し、下位mビットに"00…0₍₂₎"を出力するS-box 343₀を有し、布線によりデータmid₀₀を6系統に分岐して出力すると共に、"00…0₍₂₎"を2系統に分岐して出力する。

非線形変換部343₁'は図20Bに示すように、上位mビットにデータmid₀₁を出力し、下位mビットに"00…0₍₂₎"を出力するS-boxを有し、布線によりデータmid₀₁を6系統に分岐して出力すると共に、mビットデータ"00…0"を2系統に分岐して出力する。以下、非線形変換部343₂'～343₇'も同様に構成され、図20Cに非線形変換部343₇'の構成を示すが、その説明は省略する。これらの非線形変換部343₁'～343₇'は、それぞれデータin₁～in₇を以下のデータMID₀₁～MID₀₇に変換する。

$$MID_{01} = [mid_{01}, 00\cdots 0_{(2)}, mid_{01}, mid_{01}, mid_{01}, mid_{01}, 00\cdots 0_{(2)}] \quad (18-2)$$

$$MID_{02} = [mid_{02}, mid_{02}, 00\cdots 0_{(2)}, mid_{02}, 00\cdots 0_{(2)}, mid_{02}, mid_{02}, mid_{02}] \quad (18-3)$$

$$MID_{03} = [mid_{03}, mid_{03}, mid_{03}, 00\cdots 0_{(2)}, mid_{03}, 00\cdots 0_{(2)}, mid_{03}, mid_{03}] \quad (18-4)$$

$$mid_{11} = mid_{00} \oplus mid_{01} \oplus mid_{03} \oplus mid_{04} \oplus mid_{05} \oplus k_{i11} \quad (20-5)$$

$$mid_{15} = mid_{00} \oplus mid_{01} \oplus mid_{02} \oplus mid_{05} \oplus mid_{06} \oplus k_{i15} \quad (20-6)$$

$$mid_{16} = mid_{01} \oplus mid_{02} \oplus mid_{03} \oplus mid_{06} \oplus mid_{07} \oplus k_{i16} \quad (20-7)$$

$$mid_{17} = mid_{00} \oplus mid_{02} \oplus mid_{03} \oplus mid_{04} \oplus mid_{07} \oplus k_{i17} \quad (20-8)$$

これは図17で説明した線形変換を表す式(15-1)～(15-8)と等価な線形変換である。これにより、第4実施例と同じデータ mid_{10} , mid_{11} , mid_{12} , …, mid_{17} を生成する。なお、副鍵データ k_{ii} は8つのデータ k_{i10} , k_{i11} , k_{i12} , …, k_{i17} で構成されている。

次いで、データ mid_{10} , mid_{11} , mid_{12} , …, mid_{17} は、それぞれ図14中の非線形変換部345₀, 345₁, 345₂, …, 345₇において、データ out_0 , out_1 , out_2 , …, out_7 に非線形変換された後、結合部346において、8つのデータ out_0 , out_1 , out_2 , …, out_7 が1つのデータ Y_i^* に結合される。最後に、データ Y_i^* は、鍵データ k_{i2} による第3鍵依存線形変換部347において、例えば k_{i2} ビット左ローテーションによりデータ Y_i に線形変換され、非線形関数部304からの出力データ Y_i が生成される。

図21に示すように、第2鍵依存線形変換部344では、排他的論理和の個数が8つで図17と等価な線形変換(図17では排他的論理和の個数は24個)を実現しているため、第4実施例より高速な変換が可能になる。

また、第4実施例と同様に、非線形変換部343₀～343₇と345₀～345₇は8つずつ並列に配置されており、その非線形変換処理は相互に関連していないため、これらはすべて並列実行が可能である。さらに、非線形変換部343₀～343₇の差分確率・線形確率を p とすると、非線形関数部304全体における差分確率・線形確率は p^5 となる。

なお上述において、第2(鍵依存)線形変換部344は鍵 k_{ii} ことなく、この入力サブデータ内の排他的論理和等によってもよい。つまり図17のXOR63₀～63₇、これと対応する図18, 19, 21路部分を省略してもよい。

また、上述において、第1鍵依存線形変換部341、第2鍵依存線形変換部344及び第3鍵依存線形変換部347は必ずしも鍵依存とすることではなく、つまり、鍵データを入力すことなく、サブデータ内の線形変換を行ってもよい。

上述の第4及び第5実施例によるデータ拡散処理は、その処理手順を表すプログラムをコンピュータで実行することにより実現してもよい。その場合の処理手順は図11及び12に示すものと同様なので説明を省略する。

図22は、第1～第5実施例で説明したデータ拡散処理手順を表すプログラムを予め記録媒体に記録しておき、そのプログラムを読み出してこの発明のデータ拡散を実施する構成例を示す。中央演算装置(CPU)110、リードオンリーメモリ(ROM)120、ランダムアクセスメモリ(RAM)130、記憶装置(例えばハードディスクHD)140、I/Oインターフェース150、及びこれらを互いに接続するバス160は一般的なコンピュータ100を構成している。この発明によるデータ拡散処理を実施するプログラムは、例えば記録媒体としてのハードディスク140に格納されている。ROM120にはそれぞれのS-boxが表として格納されている。データ拡散処理を実行する際に、RAM130にそのプログラムをハードディスク140から読み込み、インターフェース150を介して平文Mが入力されるとCPU110の制御のもとにプログラムが実行され、生成された拡散データCがインターフェース150を介して出力される。

データ拡散処理を実行するプログラムは記録媒体としての任意の外部記憶装置180に格納したものを使用してもよい。その場合、ドライバ170を介して外部記憶装置180からプログラムをハードディスク140に一旦移して使用するか、RAM130に転送して使用することができる。

出力された拡散データCは、図示していないが、例えば通信回線やインターネットを介して送出されると、共通鍵を有している者のみがこの拡散データCを復号することができる。この発明により拡散されたデータCは差分解読法及び線形解読法に対し強い耐性を有しているので、より安全な情報の送信が実現できる。

ところで、前述の各実施例において、鍵生成スケジュール部を図3と同様に構成すると、データ拡散部10で k_i と k_{i+1} として使用する副鍵が、鍵スケジュール部20中の鍵拡散部21_jの出力 Q_{2j} と Q_{2j+1} (ただし $i=2j$ とする)になっていた。一方、差分解読法や線形解読法で求まる可能性の高い副鍵は k_i と k_{i-1} であるから、これらの情報を用いてデータ拡散部を組み合わせることで、他の副鍵を求めやすくなつ

ていた。

そこで、前述の各実施例を代表する図4のデータ変換装置において、以下の実施例では副鍵を生成する鍵生成スケジュール部20で実施される鍵生成アルゴリズムを、より複雑にすることでこの問題を解決する。以下の実施例では、少なくとも、副鍵 k_1 と k_{-1} が求まつても、他のデータ拡散部の出力に関する多くの情報が洩れないようにするために、図3に示した鍵拡散部22（図3中の関数 f_k ）と同様の働きをするG関数部を用いることのほかに、鍵生成についての第1の観点によれば、G関数部の出力であるL成分が記憶部に一旦記憶され、必要な個数だけL成分を求めた後に、それぞれのL成分から出来るだけ均一に必要となる情報を抽出して副鍵を生成するデータ抽出機能を備えたH関数部が設けられる。また、第2の観点によれば、G関数部の出力であるL成分からそれぞれの副鍵として使用される部分情報がH関数部で抽出され、記憶部に記憶され、必要な個数のL成分から部分情報を抽出することで副鍵が生成される。

DESの場合、主鍵のビット位置を入れ換えるだけで副鍵を生成していたので、鍵スケジュール処理は高速であった。しかし、副鍵の部分情報が知られると、直ちに主鍵の対応する情報が分かってしまう問題があった。

主鍵と副鍵の関係を複雑にするために、鍵スケジュールの処理量が大幅には増加しないように、かつ鍵スケジュール部のプログラム規模が増加しないようにするため、データ拡散部で使用するF関数あるいはF関数を構成するサブルーチン（以下ではこれらの関数を f と書く）を利用してデータ拡散関数G関数を構成して、G関数を繰り返し呼び出す、つまり繰り返し用いることで、複数個の中間値Lを生成する。

G関数は、2つの入力成分(Y, v)によって動作し、3つの出力成分(L, Y, v)を生成することとする。Y成分のビット数は主鍵Kのビット数と一致するか、それよりも大きいものとする。

G関数は、データ拡散部に副鍵を供給するために、必要な回数（M回）繰り返し呼び出されて、M個のL成分を生成する（ $0 \leq j \leq M-1$ ）。j回目に呼び出されたG関数の出力を (L_j, Y_j, v_j) と表すと、この値の一部は $(j+1)$ 回目に呼び出されたG関数の出力を $(L_{j+1}, Y_{j+1}, v_{j+1})$ と表す。

れるG関数の入力 ($Y_{j+1} = Y_j$, $v_{j+1} = v_j$) として利用される。ここで、 Y_j はその一部にKを含んだ値であり、 v_j は予め定められた値（例えば0）とする。

与えられた主鍵Kに対し、副鍵 k_i ($i = 0, 1, 2, \dots, N-1$) を以下のように定める。

$$(L_1, (Y_1, v_1)) = G(Y_0, v_0) \quad (21)$$

$$(L_{j+1}, (Y_{j+1}, v_{j+1})) = G(Y_j, v_j) \quad (j=1, 2, \dots, M-1) \quad (22)$$

$$k_i = H(i, L_1, L_2, \dots, L_M) \quad (i=0, 1, 2, \dots, N-1) \quad (23)$$

ここで、H関数は、副鍵の添字iおよび関数Gの出力であるM個のL成分を入力して、それぞれの L_j から必要に応じてiによって決められたビット位置の情報を抽出するものである。

第6の実施例

図23Aに図4で示した鍵生成スケジュール部20に適用する第6実施例の鍵生成スケジュール部の原理構成を示す。主鍵Kは中間鍵生成部220に入力され、中間鍵生成部220は縦続的に動作する複数(M段)のG関数部を有し、中間鍵 $L_1 \sim L_M$ を生成し、これら中間鍵は記憶部230に記憶される。記憶部230に記憶された中間鍵 $L_1 \sim L_M$ は副鍵生成部240でH関数部にもとづき副鍵 k_i が生成される。各部の構成作用を以下に具体的に説明する。

この例では、先に指摘した宮口らの米国特許で示されているデータランダム化部を利用するなどを想定して、先に示した図8の鍵スケジュール部の安全性を高める装置である。宮口らの米国特許で示されている鍵スケジュール部（図3）で $N=16$ の場合にこの実施例を適用する場合について説明する。

図3では8($=N/2$)段のデータ拡散部によって16個のQ成分入手する。ここでは、それぞれのQ成分を Q_j と表わす。各 Q_j は16ビットである。副鍵生成部240ではそれぞれの Q_j 成分の第1ビット目の値から副鍵 k_0 を構成し、それぞれの Q_j 成分の第2ビット目の値から副鍵 k_1 を構成し、一般にそれぞれの Q_j 成分の第iビット目の値から副鍵 k_{i-1} を構成する。すなわち、 Q_j 成分の第iビット目を $Q_j[i]$ と

表すと、副鍵 k_i は次式で示される。

$$k_{i-1} = (Q_1[i], Q_2[i], \dots, Q_j[i], \dots, Q_{16}[i]) \quad (24)$$

ここで、 $1 \leq i, j \leq 16$ に注意。

先に示した G 関数、 H 関数の枠組で、ここに述べた処理方法を見直すと以下のとおりである。ここで、 Y_j は 64 ビットの値であり、 Y_j^L は Y_j の上位 32 ビットの値、 Y_j^R は Y_j の下位 32 ビットの値を表している。

入力 (Y_j, v_j) に対する G 関数の出力を

$$(L_{j+1}, (Y_{j+1}, v_{j+1})) = G(Y_j, v_j) \quad (0 \leq j \leq 7) \quad (25)$$

と表すと、その出力結果 $(L_{j+1}, (Y_{j+1}, v_{j+1}))$ は次式により得る。

$$Y_{j+1}^L = Y_j^R \quad (26)$$

$$Y_{j+1}^R = L_{j+1} = f_k(Y_j^L, Y_j^R \oplus v_j) \quad (27)$$

$$v_{j+1} = Y_j^L \quad (28)$$

副鍵 k_i は i と $L_1 \sim L_8$ の関数として次式で示される。

$$k_{i-1} = H(i, L_1, L_2, \dots, L_8) \quad (29)$$

この H 関数は、各 L_j をビットごとに分割して $(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(32)})$ と表したとき、

副鍵 k_i を

$$k_i = (t_1^{(i)}, t_1^{(16+i)}, t_2^{(i)}, t_2^{(16+i)}, \dots, t_8^{(i)}, t_8^{(16+i)}) \quad (1 \leq i \leq 16) \quad (30)$$

と構成する。

なおこの方法では、最大 16 個の副鍵が得られるので、宮口らの米国特許で示されている暗号アルゴリズムでは、8 段の F 関数から構成した場合にまで対応できる。

図 23 A の中間鍵生成部 220 の構成を図 24 を用いて説明する。G 関数部 22-1 ～22-8 が継続的に構成され、その初段の G 関数部 22-1 に主鍵 K が Y_0 として、また定数 v_0 が入力され、各 j 段目の G 関数部 22- j は、 Y_{j-1} と v_{j-1} が入力され、 Y_{i-1} が擾乱されて、出力 L_j 、 Y_j 、 v_j を出力し、 L_j が中間鍵として出力され、 Y_j 、 v_j が次段の G 関数部 22-($j+1$) へ供給される。つまり $Y_0 = K$ 、 $v_0 = 0$ と設定した後に G 関数部 22 を 8 回呼び出す。G 関数部の構成を図 25 に示す。この構成に対し、以

以下の処理を $j = 0$ から $j = 7$ まで繰り返す。

ステップ 1 : Y_j と v_j を G 関数部 22-($j+1$) に入力すると、図 25 中のデータ分割装置 221 を用いて Y_j を 2 つのブロック (Y_j^L , Y_j^R) に分割する。

ステップ 2 : Y_j^L を v_{j+1} として出力する。また、 Y_j^L をデータ拡散部 (f_k) 222 に入力する。

ステップ 3 : Y_j^R をデータ入換器 224 に入力する。また、 Y_j^R と v_j を排他的論理和回路 223 に入力して $Y_j^R \oplus v_j$ を演算し、その結果をデータ拡散部 (f_k) 222 に入力する。

ステップ 4 : Y_j^L と $Y_j^R \oplus v_j$ を入力として受けとるとデータ拡散部 f_k 222 は、その計算結果を L_{j+1} として出力すると同時に、データ入換器 224 に入力する。

ステップ 5 : データ入換器 224 は、 Y_j^R とデータ拡散部 (f_k) 222 の計算結果 L_{j+1} を入力として受けとると、 Y_j^R を Y_{j+1}^L 、 L_{j+1} を Y_{j+1}^R とし、 $Y_{j+1} = (Y_{j+1}^L, Y_{j+1}^R)$ と連結して出力する。

G 関数部 22-1～22-8 が output した 8 個の L_j を、一旦、記憶部 230 (図 23 A) に記憶する。

次に、副鍵生成部 240 としての H 関数部の構成を図 26 を用いて説明する。H 関数部 240 は、記憶部 230 から 8 個の L 成分 L_1 ～ L_8 を読みだしたのちに以下を実行する。

ステップ 1 : 記憶部 230 から各 L_j を読みだしてビット分割器 241 に入力してそれぞれ、各 1 ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(32)}) = L_j \quad (j=1, 2, \dots, 8) \quad (31)$$

に分割する。

ステップ 2 : $(t_1^{(i)}, t_1^{(16+i)}, t_2^{(i)}, t_2^{(16+i)}, \dots, t_8^{(i)}, t_8^{(16+i)})$ をビット結合器 242 に入力して副鍵

$$k_i = (t_1^{(i)}, t_1^{(16+i)}, t_2^{(i)}, t_2^{(16+i)}, \dots, t_8^{(i)}, t_8^{(16+i)}) \quad (i=1, 2, \dots, 16) \quad (32)$$

を得る。

第 7 の実施例

上記第6実施例と同じ副鍵を出力する別の実施例を図23B、図24、図25、図27を参照して説明する。

図23Bに示すように、中間鍵生成部220で複数の中間鍵 L_j が生成される。中間鍵生成部220は図23Aのそれと同様であり、つまり図24に示したように複数のG関数部22よりなる。このG関数部22で中間鍵 L_j が生成されるごとに、その中間鍵 L_j は、副鍵生成部250で、副鍵 k_i の i と k_i のビット位置 q とにより決まるビット位置の情報が、 k_i のビット位置 q の情報 $k_{i,q}$ として選出され、記憶部260に記憶される。

つまり中間鍵生成部220と副鍵生成部250とにより以下のステップ1からステップ7までを、 $j=0$ から $j=7$ まで繰り返すことになる。

ステップ1： Y_j と v_j をG関数部22-($j+1$)に入力すると、データ分割装置221を用いて Y_j を2つのブロック(Y_j^L , Y_j^R)に分割する。

ステップ2： Y_j^L を v_{j+1} として出力する。また、 Y_j^L をデータ拡散部(f_k)222に入力する。

ステップ3： Y_j^R をデータ入換器224に入力する。また、 Y_j^R と v_j を排他的論理和回路223に入力して $Y_j^R \oplus v_j$ を入手後にデータ拡散部(f_k)222に入力する。

ステップ4： Y_j^L と $Y_j^R \oplus v_j$ を入力として受けとるとデータ拡散部(f_k)222は、その計算結果を L_{j+1} として副鍵生成部250(図23B)に入力すると同時に、データ入換器224に入力する。

ステップ5：データ入換器224は、 Y_j^R とデータ拡散部(f_k)222の計算結果 L_{j+1} を入力として受けとると、 Y_j^R を Y_{j+1}^L 、 L_{j+1} を Y_{j+1}^R とし $Y_{j+1} = (Y_{j+1}^L, Y_{j+1}^R)$ と連結して出力する。

ステップ6：副鍵生成部250は図27に示すように L_j をビット分割器251に入力して下記のように1ビットごとに

$$(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(32)}) = L_j \quad (j = 1, 2, \dots, 8) \quad (33)$$

に分割して、情報分配器252に入力する。

ステップ7：情報分配器252に入力されたビット列($t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(32)}$)はパラメータ i に対して k_i のビット位置 q により決まる L_j のビット位置の情報が、 k_i

のビット位置 q の情報とされ、

$$k_i = (t_1^{(i)}, t_1^{(16+i)}, t_2^{(i)}, t_2^{(16+i)}, \dots, t_8^{(i)}, t_8^{(16+i)}) \quad (34)$$

となるように、副鍵 k_i ごとに 16 個に分割された記憶部 260 に L_j ごとに記憶する。

ステップ 8：それぞれの k_i に 16 ビットの情報が設定されると、つまり副鍵 k_i が生成されるとその値を出力する ($i=1, 2, \dots, 16$)。

第 8 実施例

この実施例では鍵スケジュールを構成する際に、装置規模またはプログラムステップ数削減のため、暗号化で用いられている関数 f を用いる方法を説明する。

この例でも先に示した G 関数、H 関数の枠組で、説明する。

G 関数に入力 (Y_j, v_j) が与えられたときの出力を

$$(L_{j+1}, (Y_{j+1}, v_{j+1})) = G(Y_j, v_j) \quad (0 \leq j \leq 7)$$

と表し、その出力を

$$\begin{aligned} & ((Y_j^{(1)}, Y_j^{(2)}, Y_j^{(3)}, Y_j^{(4)}), v_j) \rightarrow \\ & ((L_{j+1}^{(1)}, L_{j+1}^{(2)}, L_{j+1}^{(3)}, L_{j+1}^{(4)}), [(Y_{j+1}^{(1)}, Y_{j+1}^{(2)}, Y_{j+1}^{(3)}, Y_{j+1}^{(4)}), v_{j+1}]) \end{aligned} \quad (35)$$

とする。ここで、

$$Y_{j+1}^{(i)} = f(Y_j^{(i)}) \quad (i=1, 2, 3, 4) \quad (36)$$

$$L_{j+1}^{(0)} = v_j \quad (37)$$

$$L_{j+1}^{(i)} = f(L_{j+1}^{(i-1)}) \oplus Y_{j+1}^{(i)} \quad (i=1, 2, 3, 4) \quad (38)$$

$$v_{j+1} = L_{j+1}^{(4)} \quad (39)$$

と定める。また、

$$k_i = H(i, L_1, L_2, \dots, L_8) \quad (40)$$

においては、

$$q_{i+4j} = L_{j+1}^{(i+1)} \quad (i=0, 1, 2, 3) \quad (41)$$

$$(t_i^{(0)}, t_i^{(1)}, \dots, t_i^{(7)}) = q_i \quad (i=0, 1, \dots, 31) \quad (42)$$

$$k_{(i+1)} = \left(t_{0+(i \bmod 2)}^{([i/2])}, t_{2+(i \bmod 2)}^{([i/2])}, \dots, t_{30+(i \bmod 2)}^{([i/2])} \right) \quad (i=0, 1, \dots, 15) \quad (43)$$

と定める。ただし、式(43)における $[i/2]$ は $[i/2]$ を表すものとする。

この手順を図28及び図26を使って説明する。

準備

ステップ1： T のビット幅だけ、0123456789abcdef101112... (hex) の上位から同一ビット数取りだした値を v_0 として設定する。

ステップ2：主鍵 K を Y_0 に設定する。

中間鍵生成 以下の手順を $j=0, 1, 2, \dots, 7$ について繰り返す。

ステップ1：入力 Y_j を4つに等分割し $(Y_j^{(1)}, Y_j^{(2)}, Y_j^{(3)}, Y_j^{(4)})$ とする。

ステップ2：データ拡散器611～614を用い $i=1, 2, 3, 4$ に対して、 $Y_{j+1}^{(i)} = f(Y_j^{(i)})$ をそれぞれ算出する。

ステップ3： v_j と $L_{j+1}^{(0)}$ を同一視する。

ステップ4：データ拡散器621～624を用い $i=1, 2, 3, 4$ に対して、 $f(L_{j+1}^{(i-1)})$ を演算し、その演算結果を排他的論理和回路63iに入力して $Y_{j+1}^{(i)}$ との排他的論理和演算を行って $L_{j+1}^{(i)} = f(L_{j+1}^{(i-1)}) \oplus Y_{j+1}^{(i)}$ を得る。

ステップ5： Y_{j+1} と $(Y_{j+1}^{(1)}, Y_{j+1}^{(2)}, Y_{j+1}^{(3)}, Y_{j+1}^{(4)})$ を同一視する。

ステップ6： L_{j+1} と $(L_{j+1}^{(1)}, L_{j+1}^{(2)}, L_{j+1}^{(3)}, L_{j+1}^{(4)})$ を同一視する。

ステップ7： $v_{j+1} = L_{j+1}^{(4)}$ とおく。

副鍵生成： 第6実施例と同様にして、式(43)を実現し、 k_1, k_2, \dots, k_N ($N \leq 16$)を得る。

なお、上記実施例は上記例に囚われることなく

(1) Y_0 のサイズが、 K より大きい場合は、 Y_0 の一部を K とし、残りを定数で埋める。

(2) v_0 として、任意の定数とする。

(3) それぞれの文字のビット幅を整合性が取れる範囲で任意に設定する。

(4) f を暗号化に用いるのに利用した関数以外のものを使う。

(5) H を計算する際に L_j の一部を使わない、つまり副鍵 k_i の数が少なく、 L_j の

ビット数が多い場合は、そのような状態になる。

- (6) H を計算する際に、第6実施例と同様のものを使う。
- (7) G を計算する際に第6実施例と同様のものを使う。
- (8) 第7実施例と同様にして、中間鍵の全てを生成することなく1つの中間鍵が生成されるごとに記憶部260の k_i の対応ビット位置に計算結果を格納する。などとしても実行可能である。

中間鍵生成部220、副鍵生成部240、250は図22に示したコンピュータによりプログラムを読み出し実行するようにしてもよい。

発明の効果

以上、詳細に説明したように、この発明によれば、データを秘匿するための暗号化装置に用いるためのデータ変換装置について、安全性と高速性を同時に満たすような構造にすることによって、段数を大幅に増加させることなく安全性を確保し、かつ高速な暗号化処理が可能となるようなデータ変換装置を提供することができる。これにより、秘密鍵の制御のもとでデータをブロック単位で暗号化または復号を行う共通鍵暗号方式による暗号化装置に有用である。

また、この発明による鍵生成スケジュールによれば、第6および第7の実施例では、 $k_6, k_7, k_8, k_9, k_{10}, k_{11}$ が判明した場合でも、それぞれの L_j 成分の12ビット分（例えば、第6, 7, 8, 9, 10, 11, 22, 23, 24, 25, 26, 27ビット目）が求まっただけであるから、DESおよび宮口らの米国特許で示されている鍵スケジュール部の安全性に関する問題は解決されている。

付録

表記

$\#\{a | cond\}$: 条件condを満たす集合aの個数。

$\Delta x, \Gamma x$: x の差分値、 x のマスク値。

$a \bullet b$: a と b のビット毎の論理積に対する偶数パリティ値。

線形変換部 : $P = [t_{ij}]$, $t_{ij} \in \{0, 1\}$, $0 \leq i, j \leq n-1$

線形変換部の入力 : $z = [z_0, \dots, z_{n-1}]$, $z_i \in GF(2^n)$, $0 \leq i \leq n-1$

線形変換部の出力 : $z' = [z'_0, \dots, z'_{n-1}] = P_z$, $z'_i \in GF(2^n)$, $0 \leq i \leq n-1$

定義

[定義 1] Δx , Δy , Γx , Γy が与えられたとき、s-box の差分確率 $DP^s(\Delta x \rightarrow \Delta y)$ と線形確率 $LP^s(\Gamma y \rightarrow \Gamma x)$ を以下のように定める：

$$DP^s(\Delta x \rightarrow \Delta y) = \#\{x \in GF(2^n) \mid s(x) + s(x + \Delta x) = \Delta y\} / 2^n$$

$$LP^s(\Gamma y \rightarrow \Gamma x) = (2 \times \#\{x \in GF(2^n) \mid x \bullet \Gamma x = s(x) \bullet \Gamma y\} / 2^n - 1)^2$$

[定義 2] s-box の最大差分確率 p_s と最大線形確率 q_s を以下のように定める。

$$p_s = \max_{\Delta x \neq 0, \Delta y} DP^s(\Delta x \rightarrow \Delta y)$$

$$q_s = \max_{\Gamma x, \Gamma y \neq 0} LP^s(\Gamma y \rightarrow \Gamma x)$$

[定義 3] 差分解読法では入力差分値 Δx が非零である s-box 、線形解読法では出力マスク値 Γy が非零である s-box のことを active s-box と呼ぶ。

[定義 4] 2-round SPN構造で構成されるラウンド関数において、 Δx , Δy , Γx , Γy が与えられたとき、差分確率 $p(\Delta x \rightarrow \Delta y)$ と線形確率 $q(\Gamma y \rightarrow \Gamma x)$ を以下のように定める。

$$p(\Delta x \rightarrow \Delta y) = \max_{\Delta z} \prod_{i=0}^{n-1} DP^s(\Delta x_i \rightarrow \Delta z_i) p(\Delta z \rightarrow \Delta z'_i) DP^s(\Delta z'_i \rightarrow \Delta y_i)$$

$$q(\Gamma y \rightarrow \Gamma x) = \max_{\Gamma z'} \prod_{i=0}^{n-1} LP^s(\Gamma y_i \rightarrow \Gamma z'_i) p(\Gamma z' \rightarrow \Gamma z_i) LP^s(\Gamma z_i \rightarrow \Gamma x_i)$$

ここで、 $\Delta x = (\Delta x_0, \dots, \Delta x_{n-1})$ であり、 Δy , Γx , Γy も同様である。更に、 $p(\Delta z \rightarrow \Delta z'_i)$ は、各 i に対して、 Δz が線形変換部によって $\Delta z'_i$ に変換されることを表しているものとする。即ち、全ての i に対して、 $\exists \Delta z'_i$, s. t. $p(\Delta z \rightarrow \Delta z'_i) = 1$ である。 $p(\Gamma z' \rightarrow \Gamma z_i)$ も同様である。

[定義 5] 2-round SPN構造で構成されるラウンド関数の最大差分確率 p と最大線形確率 q を以下のように定める。

$$p = \max_{\Delta x \neq 0, \Delta y} p(\Delta x \rightarrow \Delta y)$$

$$q = \max_{\Gamma x, \Gamma y \neq 0} q(\Gamma y \rightarrow \Gamma x)$$

[定義 6] 2-round SPN構造に対して、全てのs-box が全単射である時、差分解読法と線形解読法におけるactive s-box の最少個数 n_d 、 n_l は以下のように与えられる。

$$n_d = \min_{\Delta z \neq 0} [H_w(\Delta z) + H_w(\Delta z')]$$

$$n_l = \min_{\Gamma z' \neq 0} [H_w(\Gamma z) + H_w(\Gamma z')]$$

ここで、

$$H_w(z) = \#\{0 \leq i < n \mid z_i \neq 0\}$$

とする。また、 Δz 、 Γz は線形変換部の入力差分値と入力マスク値を表し、 $\Delta z'$ 、 $\Gamma z'$ は線形変換部の出力差分値と出力マスク値を表す。

[定理 1] 2-round SPN構造のラウンド関数における最大差分確率 p と最大線形確率 q は以下の関係式を満たす。

$$p \leq p_s^{n_d} \text{ 及び } q \leq q_s^{n_l}$$

[Proof sketch]

差分解読法の場合について考える。定義 2 と定義 3 より、active s-box における差分確率は p_s 以下である。従って、ある Δx 、 Δy におけるactive s-box の個数を n_{dz} とすると、そのときのラウンド関数の差分確率は、定義 4 より、 $p_s^{n_{dz}}$ 以下となる。一方、 n_d をactive s-box の最少個数とすると、全ての Δx 、 Δy に対して $n_d \leq n_{dz}$ 、 $p_s \leq 1$ であるから、

$$\forall n_{dz} p_s^{n_{dz}} \leq p_s^{n_d}$$

となる。従って、このラウンド関数における最大差分確率 p は、 $p \leq p_s^{n_d}$ である。

同様のことが、線形解読法の場合についてもいえ、 $q \leq q_s^{n_l}$ が導かれる。

[定理 2] Concatenation Rules

$X \vdash (Y, Z)$ を、データ X がデータ Y とデータ Z の 2 つのデータに同じく複製されることを表すとする。即ち、 $X = Y = Z$ が成立する。このとき、以下の関係が成立

する：

$$X = Y \oplus Z \Rightarrow \Delta X = \Delta Y \oplus \Delta Z, \quad \Gamma X = \Gamma Y = \Gamma Z \quad (\text{XOR operation})$$

$$X \vdash (Y, Z) \Rightarrow \Delta X = \Delta Y = \Delta Z, \quad \Gamma X = \Gamma Y \oplus \Gamma Z \quad (\text{Branch operation})$$

[差分解読法及び線形解読法に対する耐性]

定理1の式で示したように、差分解読法及び線形解読法に対するラウンド関数の耐性は、active s-box の最少個数 n_d , n_l から決めることができる。例えば、行列 P_E が表す線形変換は全単射であるので、 $H_s(\Delta z) \geq 2$ ならば $H_s(\Delta z') \geq 1$ であり、また $H_s(\Delta z) = 1$ ならば $H_s(\Delta z') \geq 3$ であることから、active s-box の個数が 2 になることはあり得ないので、 $n_d \geq 3$ となる。ところで、例えば、 $\Delta z_0 = \Delta z_2 \neq 0$, $\Delta z_1 = \Delta z_3 = 0$ の時、 $\Delta z'_0 = \Delta z'_2 \neq 0$, $\Delta z'_1 = \Delta z'_2 = \Delta z'_3 = 0$ ($H_s(\Delta z') = 1$) となり、active s-box の個数は、 $H_s(\Delta z) + H_s(\Delta z') = 3$ である。従って、 $n_d = 3$ となるので、定理1よりこのラウンド関数の耐性は $p \leq p_s^3$ である。また n_l についても同様の議論ができる、 $n_l = 3$, $q \leq q_s^3$ であることが示せる。

請求の範囲

1. 複数の鍵データを保持する鍵記憶手段と、上記複数の鍵データが与えられ、それぞれ鍵依存の非線形変換を行う非線形関数部をそれぞれ含む縦続接続された複数のラウンド処理部とを含み、入力データを鍵データに依存して別のデータに変換するデータ変換装置であり、各上記ラウンド処理部の上記非線形関数部は：

上記ラウンド処理部への入力データを、上記鍵記憶手段に蓄積された第1鍵データに基づいて線形変換を行う第1鍵依存線形変換手段と、

上記第1鍵依存線形変換手段の出力データをn個のサブデータに分割する分割手段と、nは4以上の正数であり、

上記n個のサブデータのそれぞれに非線形変換を行う第1非線形変換手段と、

上記鍵記憶手段に蓄積された第2鍵データと、上記第1非線形変換手段の各々の出力サブデータとで線形変換を行う第2鍵依存線形変換手段と、

上記第2鍵依存線形変換手段のn個の出力サブデータのそれぞれに非線形変換を行う第2非線形変換手段と、及び

上記第2非線形変換手段のn個の出力サブデータを結合して上記非線形関数手段の出力とする結合手段、

とを含み、上記第2鍵依存線形変換手段は、その入力に対し $n \times n$ 行列で規定される排他的論理和を行う線形変換層を含んでいる。

2. 請求項1記載のデータ変換装置において、

上記入力データを二つの部分データに分割する初期分割手段と、

上記部分データの一つを入力とする上記非線形関数手段と、

上記非線形関数手段の出力データを残りの部分データの一つに作用させる線形演算手段と、

二つの部分データを一つの出力データに結合する最終結合手段、

とを備える。

3. 請求項2に記載のデータ変換装置において、

上記入力データに変換を行って上記初期分割手段へ供給する初期変換手段を備えることを特徴とするデータ変換装置。

4. 請求項 2 又は 3 に記載のデータ変換装置において、上記最終結合手段の出力データに変換を行って上記データ変換装置の出力データとする最終変換手段を備えることを特徴とするデータ変換装置。
5. 請求項 3 又は 4 に記載のデータ変換装置において、上記初期変換手段及び上記最終変換手段の少くとも一方は、上記鍵記憶手段に蓄積された鍵データに基づいて変換を行う鍵依存変換手段であることを特徴とするデータ変換装置。
6. 請求項 1 乃至 5 のいずれか 1 つに記載のデータ変換装置において、上記非線形関数手段は、上記鍵記憶手段に蓄積された第 3 鍵データにより上記結合手段の出力データを線形変換して上記非線形関数手段の出力とする第 3 鍵依存線形変換手段を備えることを特徴とするデータ変換装置。
7. 請求項 1 乃至 6 のいずれか 1 つに記載のデータ変換装置において、上記第 1 鍵依存線形変換手段、上記第 2 鍵依存線形変換手段、上記第 3 鍵依存線形手段のいずれかは、固定された線形変換を行う線形変換手段であることを特徴とするデータ変換装置。
8. 請求項 1 乃至 7 のいずれか 1 つに記載のデータ変換装置において、上記第 1 非線形変換手段及び上記第 2 非線形変換手段は、その各入力サブデータを、二つのサブブロックに分割する手段と、その分割された二つのサブブロックに対し、それぞれ線形変換を行う手段と、非線形変換を行う手段とを縦統的に行う手段と、その縦統的に行う手段の各変換出力サブブロックを統合して対応入力サブデータに対する非線形変換出力サブデータとする手段を備えていることを特徴とするデータ変換装置。
9. 請求項 1 乃至 8 のいずれか 1 つに記載のデータ変換装置において、上記 $n \times n$ 行列は、予め決定した安全性閾値 T に対し、それがハミング重み $T-1$ 以上となる n 個の列ベクトルから構成されている。
10. 請求項 9 に記載のデータ変換装置において、上記行列は、上記ハミング重みが $T-1$ 以上となる列ベクトルから構成された複数の行列の候補から、最大の n を与えるものとして選択されたものであり、上記 n は active s-box の最少個数である。

1 1. 請求項 1 乃至 1 0 のいずれか 1 つに記載のデータ変換装置において、上記 $n \times n$ 行列は 4×4 行列である。

1 2. 請求項 1 1 に記載のデータ変換装置において、

上記第 2 線形変換手段は、上記第 1 非線形変換手段よりの 4 つの出力 A1, A2, A3, A4 を入力して、

$$B1 = A1 \oplus A3 \oplus A4$$

$$B2 = A2 \oplus A3 \oplus A4$$

$$B3 = A1 \oplus A2 \oplus A3$$

$$B4 = A1 \oplus A2 \oplus A4$$

をそれぞれ演算して、データ B1, B2, B3, B4 を出力する手段であることを特徴とするデータ変換装置。

1 3. 請求項 1 2 に記載のデータ変換装置において、

上記第 2 線形変換手段は鍵依存線形変換手段であって、上記鍵記憶手段よりの鍵データ $k2 = [k21, k22, k23, k24]$ も入力され、上記 B1, B2, B3, B4 を得る演算に、それぞれ $k21, k22, k23, k24$ も排他的論理和算されることを特徴とするデータ変換装置。

1 4. 請求項 1 1 に記載のデータ変換装置において、

上記第 1 非線形変換手段は上記分割手段よりの 4 個の各 m ビットのサブデータ i in1, in2, in3, in4 の in1 を $4m$ ビットの $MI1 = [A1, 00 \cdots 0_{(2)}, A1, A1]$ に変換する手段と、in2 を $4m$ ビットの $MI2 = [00 \cdots 0_{(2)}, A2, A2, A2]$ に変換する手段と、in3 を $4m$ ビットの $MI3 = [A3, A3, A3, 00 \cdots 0_{(2)}]$ に変換する手段と、in4 を $4m$ ビットの $MI4 = [A4, A4, A4, 00 \cdots 0_{(2)}, A4]$ に変換する手段とよりなり、

上記第 2 線形変換手段は、上記第 1 非線形変換手段よりのデータ MI1, MI2, MI3, MI4 を入力して、 $B = MI1 \oplus MI2 \oplus MI3 \oplus MI4$ を演算し、 $B = [B1, B2, B3, B4]$ を出力する手段であることを特徴とするデータ変換装置。

1 5. 請求項 1 4 に記載のデータ変換装置において、

上記第 2 線形変換手段は鍵依存線形変換手段であって、上記鍵記憶手段よりの $4m$ ビットの鍵データ $k2$ も入力され、上記 B の演算に $k2$ も排他的論理和算される

ことを特徴とするデータ変換装置。

16. 請求項1乃至10のいずれか1つに記載のデータ変換装置において、上記 $n \times n$ 行列は 8×8 行列である。

17. 請求項16に記載のデータ変換装置において、

上記第2線形変換手段は、その8つの出力サブデータB1～B8を、上記第1非線形変換手段の8つの出力サブデータA1, A2, …, A8のうち、6つのサブデータの排他的論理和算により4つの出力サブデータB1, B2, B3, B4を得、上記8つの出力サブデータ中の5つのサブデータの排他的論理和算により4つの出力サブデータB5, B6, B7, B8を得る手段であることを特徴とするデータ変換装置。

18. 請求項17に記載のデータ変換装置において、上記第2線形変換手段は鍵依存線形変換手段であって、上記鍵記憶手段に蓄積された鍵データk2=[k21, k22, k23, k24, K25, k26, k27, k28]も入力され、上記出力サブデータ[B1, B2, B3, B4, B5, B6, B7, B8]を得る排他的論理和算にそれぞれ、鍵データk21, k22, k23, k24, k25, k26, k27, k28が排他的論理和されることを特徴とするデータ変換装置。

19. 請求項16に記載のデータ変換装置において、

上記第1非線形変換手段は、上記分割手段よりの8個の各mビットのサブデータin1～in8をそれぞれ8mビットのデータ

$$MI1 = [00\cdots0_{(2)}, A1, A1, A1, A1, A1, 00\cdots0_{(2)}, A1],$$

$$MI2 = [A2, 00\cdots0_{(2)}, A2, A2, A2, A2, A2, 00\cdots0_{(2)}],$$

$$MI3 = [A3, A3, 00\cdots0_{(2)}, A3, 00\cdots0_{(2)}, A3, A3, A3],$$

$$MI4 = [A4, A4, A4, 00\cdots0_{(2)}, A4, 00\cdots0_{(2)}, A4, A4],$$

$$MI5 = [A5, 00\cdots0_{(2)}, A5, A5, A5, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A5],$$

$$MI6 = [A6, A6, 00\cdots0_{(2)}, A6, A6, A6, 00\cdots0_{(2)}, 00\cdots0_{(2)}],$$

$$MI7 = [A7, A7, A7, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A7, A7, 00\cdots0_{(2)}], \text{及び}$$

$$MI8 = [00\cdots0_{(2)}, A8, A8, A8, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A8, A8]$$

に変換する手段であり、

上記第2線形変換手段は、上記第1非線形変換手段よりのデータMI1～MI8を入

力して、 $B = MI1 \oplus MI2 \oplus MI3 \oplus MI4 \oplus MI5 \oplus MI6 \oplus MI7 \oplus MI8$ を演算し、 $B = [B1, B2, B3, B4, B5, B6, B7, B8]$ を出力する手段であることを特徴とするデータ変換装置。

20. 請求項19に記載のデータ変換装置において、

上記第2線形変換手段は鍵依存線形変換手段であって、上記鍵記憶手段の8mビットの鍵データ $k2$ も入力され、上記 B の演算に $k2$ も排他的論理和算されることを特徴とするデータ変換装置。

21. 鍵記憶手段に保持された複数の鍵データが与えられ、それぞれ鍵依存の非線形変換を行う非線形関数処理を含むラウンド処理を複数回継続して実行することにより、入力データを鍵データに依存して別のデータに変換するデータ変換プログラムが記録された記録媒体であり、各上記ラウンド処理の上記非線形関数処理は：

上記ラウンド処理部への入力データを、上記鍵記憶手段に保持された第1鍵データに基づいて線形変換を行う第1鍵依存線形変換ステップと、

上記第1鍵依存線形変換ステップによる出力データを n 個のサブデータに分割する分割ステップと、 n は4以上の整数であり、

上記 n 個のサブデータのそれぞれに非線形変換を行う第1非線形変換ステップと、

第2鍵データと、上記第1非線形変換ステップの出力サブデータとで線形変換を行う第2鍵依存線形変換ステップと、

上記第2鍵依存線形変換ステップによる n 個の出力サブデータのそれぞれに第2の非線形変換を行う第2非線形変換ステップと、及び

上記第2非線形変換ステップによる n 個の出力サブデータを結合して上記非線形関数処理結果として出力する結合ステップ、

とを含み、

上記第2鍵依存線形変換ステップは、その入力に対し $n \times n$ 行列で規定される排他的論理和を行う排他的論理和線形変換ステップを含んでいる。

22. 請求項21に記載の記録媒体において、上記データ変換プログラムは、

上記入力データを二つの部分データに分割する初期分割ステップと、

上記部分データの一つを入力とする上記非線形関数処理を行うステップと、
上記非線形関数処理ステップの出力データを残りの部分データの一つに作用さ
せる線形演算ステップと、
二つの部分データを一つの出力データに結合する最終結合ステップ、
とを含む。

23. 請求項22に記載の記録媒体において、上記データ変換プログラムは、上記入力データに変換を行って上記初期分割ステップへ供給する初期変換ステップを含む。

24. 請求項22又は23に記載の記録媒体において、上記データ変換プログラムは 上記最終結合ステップの出力データに変換を行って上記データ変換出力データとする最終変換ステップを含む。

25. 請求項23又は24に記載の記録媒体において、上記データ変換プログラムの、上記初期変換ステップ及び上記最終変換ステップの少なくとも一方は、鍵データに基づいて変換を行う鍵依存変換ステップである。

26. 請求項21乃至25のいずれか1つに記載の記録媒体において、
上記非線形関数処理ステップは、上記鍵保持手段に保持された第3鍵データに
より上記結合ステップの出力データを線形変換して上記非線形関数処理ステップ
の出力とする第3鍵依存線形変換ステップを含む。

27. 請求項21乃至28のいずれか1つに記載の記録媒体において、
上記第1鍵依存線形変換ステップ、上記第2鍵依存線形変換ステップ、上記第
3鍵依存線形変換ステップのいずれかは、固定された線形変換を行う線形変換ス
テップである。

28. 請求項21乃至27のいずれか1つに記載の記録媒体において、
上記第1非線形変換ステップ及び上記第2非線形変換ステップは、その各入力
サブデータを、二つのサブブロックに分割するステップと、その分割された二つ
のサブブロックに対し、それぞれ線形変換を行う処理と、非線形変換を行う処理
とを縦続的に行うステップと、その縦続的に行うステップの各変換出力サブブロ
ックを統合して対応入力サブデータに対する非線形変換出力サブデータとするス

ステップを含む。

29. 請求項 21 乃至 28 のいずれか 1 つに記載の記録媒体において、上記 $n \times n$ 行列は、予め決定した安全性閾値 T に対し、それぞれがハミング重み $T-1$ 以上となる n 個の列ベクトルから構成されている。

30. 請求項 29 に記載の記録媒体において、上記行列は、上記ハミング重みが $T-1$ 以上となる列ベクトルから構成された複数の行列の候補から、最大の n_s を与えるものとして選択されたものであり、上記 n_s は active s-box の最少個数である。

31. 請求項 21 から 30 のいずれかに記載の記録媒体において、上記 $n \times n$ 行列は 4×4 行列である。

32. 請求項 31 に記載の記録媒体において、

上記第 2 線形変換ステップは、上記第 1 非線形変換ステップよりの 4 つの出力 A_1, A_2, A_3, A_4 を入力して、

$$B_1 = A_1 \oplus A_3 \oplus A_4$$

$$B_2 = A_2 \oplus A_3 \oplus A_4$$

$$B_3 = A_1 \oplus A_2 \oplus A_3$$

$$B_4 = A_1 \oplus A_2 \oplus A_4$$

をそれぞれ演算して、データ B_1, B_2, B_3, B_4 を出力するステップである。

33. 請求項 32 に記載の記録媒体において、

上記第 2 線形変換ステップは鍵依存線形変換ステップであって、上記鍵記憶手段よりの鍵データ $k_2 = [k_{21}, k_{22}, k_{23}, k_{24}]$ も入力され、上記 B_1, B_2, B_3, B_4 を得る演算に、それぞれ $k_{21}, k_{22}, k_{23}, k_{24}$ も排他的論理和算される。

34. 請求項 32 又は 33 に記載の記録媒体において、

上記第 1 非線形変換ステップは上記分割ステップよりの 4 個の各 m ビットのサブデータ in_1, in_2, in_3, in_4 の in_1 を $4m$ ビットの $MI1 = [A_1, 00 \cdots 0_{(2)}, A_1, A_1]$ に変換するステップと、 in_2 を $4m$ ビットの $MI2 = [00 \cdots 0_{(2)}, A_2, A_2, A_2]$ に変換するステップと、 in_3 を $4m$ ビットの $MI3 = [A_3, A_3, A_3, 00 \cdots 0_{(2)}]$ に変換するステップと、 in_4 を $4m$ ビットの $MI4 = [A_4, A_4, 00 \cdots 0_{(2)}, A_4]$ に変換するステップとよりなり、

上記第2線形変換ステップは、上記第1非線形変換ステップよりのデータMI1, MI2, MI3, MI4を入力して、 $B = MI1 \oplus MI2 \oplus MI3 \oplus MI4$ を演算し、 $B = [B1, B2, B3, B4]$ を出力するステップである。

35. 請求項34に記載の記録媒体において、

上記第2線形変換ステップは鍵依存線形変換ステップであって、上記鍵記憶手段よりの4mビットの鍵データk2も入力され、上記Bの演算にk2も排他的論理和算される。

36. 請求項21乃至30のいずれか1つに記載の記録媒体において、上記n × n行列は8 × 8行列である。

37. 請求項36に記載の記録媒体において、

上記第2線形変換ステップは、その8つの出力サブデータB1～B8を、上記第1非線形変換ステップの8つの出力サブデータA1, A2, …, A8のうち、6つのサブデータの排他的論理和算により4つの出力サブデータB1, B2, B3, B4を得、上記8つの出力サブデータ中の5つのサブデータの排他的論理和算により4つの出力サブデータB5, B6, B7, B8を得るステップである。

38. 請求項37に記載の記録媒体において、上記第2線形変換ステップは鍵依存線形変換ステップであって、上記鍵記憶手段に蓄積された鍵データ $k2 = [k21, k22, k23, k24, k25, k26, k27, k28]$ も入力され、上記出力サブデータ[B1, B2, B3, B4, B5, B6, B7, B8]を得る排他的論理和算にそれぞれ、鍵データk21, k22, k23, k24, k25, k26, k27, k28が排他的論理和される。

39. 請求項37又は38に記載の記録媒体において、

上記第1非線形変換ステップは、上記分割ステップよりの8個の各mビットのサブデータin1～in8をそれぞれ8mビットのデータ

$$MI1 = [00\cdots0_{(2)}, A1, A1, A1, A1, A1, 00\cdots0_{(2)}, A1],$$

$$MI2 = [A2, 00\cdots0_{(2)}, A2, A2, A2, A2, A2, 00\cdots0_{(2)}],$$

$$MI3 = [A3, A3, 00\cdots0_{(2)}, A3, 00\cdots0_{(2)}, A3, A3, A3],$$

$$MI4 = [A4, A4, A4, 00\cdots0_{(2)}, A4, 00\cdots0_{(2)}, A4, A4],$$

$$MI5 = [A5, 00\cdots0_{(2)}, A5, A5, A5, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A5].$$

$MI6 = [A6, A6, 00\cdots0_{(2)}, A6, A6, A6, 00\cdots0_{(2)}, 00\cdots0_{(2)}]$ 、
 $MI7 = [A7, A7, A7, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A7, A7, 00\cdots0_{(2)}]$ 、及び
 $MI8 = [00\cdots0_{(2)}, A8, A8, A8, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A8, A8]$

に変換するステップであり、

上記第2線形変換ステップは、上記第1非線形変換ステップよりのデータ $MI1\sim MI8$ を入力して、

$$B = MI1 \oplus MI2 \oplus MI3 \oplus MI4 \oplus MI5 \oplus MI6 \oplus MI7 \oplus MI8$$

を演算し、 $B = [B1, B2, B3, B4, B5, B6, B7, B8]$ を出力するステップである。

4 0. 請求項3 9に記載の記録媒体において、

上記第2線形変換ステップは鍵依存線形変換ステップであって、上記鍵記憶手段の8mビットの鍵データ $k2$ も入力され、上記Bの演算に $k2$ も排他的論理和算される。

4 1. 請求項1乃至2 0のいずれか1つのデータ変換装置において、更に、

主鍵 K が入力され、継続的に動作するM段からなり、各段より中間値 L_{j+1} ($j=0, 1, \dots, M-1$) 成分を生成するG関数手段と、

そのG関数手段の出力である各中間値 L_j 成分を一旦記憶しておくための中間値記憶手段と、

複数個の L_j 成分からN個の副鍵を生成し、上記鍵記憶手段に上記複数の鍵データとして記憶する部分情報抽出機能を備えたH関数手段、
とを含み、

上記G関数手段は、上記主鍵 K を、 Y_0 の少なくとも一部として取り込み、 $j+1$ 段目が ($j=0, 1, \dots, M-1$) j 段目の出力 (L_j, Y_j, v_j) 中の Y_j と v_j を入力し、
その入力を拡散して、 $L_{j+1}, Y_{j+1}, v_{j+1}$ を出力し、を K とする手段であり、

上記H関数手段は、 i ($i=1, 2, \dots, N$) と上記中間値記憶手段の L_1, L_2, \dots, L_M とを入力とし、 i によって決められたビット位置の情報を L_1, \dots, L_M より抽出する副鍵 k_i を出力し、上記副鍵は上記複数の鍵データとして上記鍵記憶手段に蓄積される。

4 2. 請求項1乃至2 0のいずれか1つに記載のデータ変換装置において、更

に、

主鍵Kが入力され、継続的に動作するM段からなり、各段より中間値L_j (j=0, 1, …, M-1) 成分を生成するG関数手段と、

そのG関数手段で生成された複数個のL_j 成分から副鍵を生成する部分情報抽出機能を備えたH関数手段と、

そのH関数手段の出力を副鍵k_i に対応する値として記憶しておくための中間値記憶手段、

とを含み、

上記G関数手段は、上記主鍵Kを、Y₀ の少なくとも一部として取り込み、j+1段目がj段目の出力 (L_j, Y_j, v_j) 中のY_j, v_jを入力し、その入力を拡散してL_{j+1}, Y_{j+1}, v_{j+1} を出力する手段であり、

上記H関数手段は、i, q, L_j (1 ≤ i ≤ N, 1 ≤ j ≤ M, 1 ≤ q ≤ k_i のビット数) を入力として、L_j から、iとqによって決められたビット位置情報を抽出して、副鍵k_i のビット位置qの情報とする手段であり、上記副鍵は上記複数の鍵データとして上記鍵記憶手段に記憶される。

4 3. 請求項4 1又は4 2に記載のデータ変換装置において、上記G関数手段は、入力されたY_jを2つのブロック (Y_j^L, Y_j^R) に分割し、Y_j^Lをv_{j+1}として出力するデータ分割手段と、

上記Y_j^Rと上記v_jとからY_j^R⊕v_jを演算する排他的論理和手段と、

上記Y_j^Lと上記排他的論理和手段の出力とがあたえられ、それらを互いに拡散した結果をL_{j+1}として出力するデータ拡散手段と、

上記Y_j^RをY_{j+1}^Lとし、上記L_{j+1}をY_{j+1}^Rとし、これらY_{j+1}^LとY_{j+1}^Rを互いに連結してY_{j+1}=(Y_{j+1}^L, Y_{j+1}^R) として出力するデータ入れ換え手段、

とを含む。

4 4. 請求項4 1に記載のデータ変換装置において、上記H関数手段は、

上記中間値記憶手段から読み出した各L_jを各1ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(2^n)}) = L_j \quad (j=1, 2, \dots, M)$$

に分割するビット分割手段と、

得られた $(t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \dots, t_M^{(i)}, t_M^{(N+i)})$ を結合して副鍵

$$k_i = (t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \dots, t_M^{(i)}, t_M^{(N+i)})$$

$$(i=1, 2, \dots, N)$$

を出力するビット結合手段、

とを含む。

4 5. 請求項 4 2 に記載のデータ変換装置において、上記 H 関数手段は、
上記各 L_j を各 1 ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(2N)}) = L_j \quad (j=1, 2, \dots, M)$$

に分割するビット分割手段と、

上記ビット $(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(2N)})$ から i に対して k_i のビット位置 q により決ま
る L_j のビット位置の情報が、 k_i のビット位置となるよう結合して副鍵

$$k_i = (t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \dots, t_M^{(i)}, t_M^{(N+i)})$$

$$(i=1, 2, \dots, N)$$

を出力するビット結合手段、

とを含む。

4 6. 請求項 4 1 又は 4 2 に記載のデータ変換装置において、

上記 G 関数手段は次の演算処理、

$(L_{j+1}, (Y_{j+1}, v_{j+1})) = G(Y_j, v_j) \quad (0 \leq j \leq M-1)$ において、出力結果を

$((Y_j^{(1)}, Y_j^{(2)}, Y_j^{(3)}, v_j) \rightarrow$

$$\left((L_{j+1}^{(1)}, L_{j+1}^{(2)}, L_{j+1}^{(3)}, L_{j+1}^{(4)}), [(Y_{j+1}^{(1)}, Y_{j+1}^{(2)}, Y_{j+1}^{(3)}, Y_{j+1}^{(4)}), v_{j+1}] \right)$$

ここで、 $Y_{j+1}^{(i)} = f(Y_j^{(i)}) \quad (i=1, 2, 3, 4)$

$$L_{j+1}^{(0)} = v_j$$

$$L_{j+1}^{(i)} = f(L_{j+1}^{(i-1)}) \oplus Y_{j+1}^{(i)} \quad (i=1, 2, 3, 4)$$

$$v_{j+1} = L_{j+1}^{(4)}$$

を実行する手段であり、

上記 H 関数手段は次の演算処理

$$k_i = H(i, L_1, L_2, \dots, L_M)$$

において、

$$q_{4i+j} = L_{j+1}^{(i+1)} \quad (i=0, 1, 2, 3)$$

$$(t_i^{(0)}, t_i^{(1)}, \dots, t_i^{(7)}) = q_i \quad (i=0, 1, \dots, 31)$$

$$k_{(i+1)} = \left(t_{0+(i \bmod 2)}^{([i/2])}, t_{2+(i \bmod 2)}^{([i/2])}, \dots, t_{30+(i \bmod 2)}^{([i/2])} \right) \quad (i=0, 1, \dots, N-1)$$

を実行する手段である。

4.7. 主鍵から副鍵をスケジュールする装置であって、

主鍵Kが入力され、縦続的に動作するM段からなり、各段より中間値L_j (j=0,

1, …, M-1) 成分を生成するG関数手段と、

そのG関数手段の出力である各中間値L_j 成分を一旦記憶しておくための中間値記憶手段と、

複数個のL_j 成分からN個の副鍵を生成する部分情報抽出機能を備えたH関数手段、

とを含み、

上記G関数手段は、上記主鍵Kを、Y₀ の少なくとも一部として取り込み、j+1段目が (j=0, 1, …, M-1) j段目の出力 (L_j, Y_j, v_j) 中のY_j とv_j を入力し、その入力を拡散して、L_{j+1}, Y_{j+1}, v_{j+1} を出力し、をKとする手段であり、

上記H関数手段は、i (i=1, 2, …, N) と上記中間値記憶手段のL₁, L₂, …, L_N とを入力とし、iによって決められたビット位置の情報をL₁, …, L_N より抽出する副鍵k_i を出力する手段であることを特徴とする暗号鍵スケジュール装置。

4.8. 主鍵から副鍵をスケジュールする装置であって、

主鍵Kが入力され、縦続的に動作するM段からなり、各段より中間値L_j (j=0,

1, …, M-1) 成分を生成するG関数手段と、

そのG関数手段で生成された複数個のL_j 成分から副鍵を生成する部分情報抽出機能を備えたH関数手段と、

そのH関数手段の出力を副鍵k_i に対応する値として記憶しておくための中間値記憶手段、

とを含み、

上記G関数手段は、上記主鍵Kを、 Y_0 の少なくとも一部として取り込み、 $j+1$ 段目が j 段目の出力 (L_j, Y_j, v_j) 中の Y_j, v_j を入力し、その入力を拡散して $L_{j-1}, Y_{j+1}, v_{j+1}$ を出力する手段であり、

上記H関数手段は、 i, q, L_j ($1 \leq i \leq N, 1 \leq j \leq M, 1 \leq q \leq k_i$ のビット数) を入力として、 L_j から、 i と q によって決められたビット位置情報を抽出して、副鍵 k_i のビット位置 q の情報とする手段であることを特徴とする暗号鍵スケジュール装置。

4 9. 請求項4 7又は4 8に記載の装置において、上記G関数手段は、
入力された Y_j を2つのブロック (Y_j^L, Y_j^R)に分割し、 Y_j^L を v_{j+1} として出力するデータ分割手段と、

上記 Y_j^R と上記 v_j とから $Y_j^R \oplus v_j$ を演算する排他的論理手段と、

上記 Y_j^L と上記排他的論理手段の出力とがあたえられ、それらを互いに拡散した結果を L_{j+1} として出力するデータ拡散手段と、

上記 Y_j^R を Y_{j+1}^L とし、上記 L_{j+1} を Y_{j+1}^R とし、これら Y_{j+1}^L と Y_{j+1}^R を互いに連結して $Y_{j+1} = (Y_{j+1}^L, Y_{j+1}^R)$ として出力するデータ入れ換え手段、

とを含む。

5 0. 請求項4 7に記載の装置において、上記H関数手段は、

上記中間値記憶手段から読み出した各 L_j を各1ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(2N)}) = L_j \quad (j=1, 2, \dots, M)$$

に分割するビット分割手段と、

得られた $(t_1^{(1)}, t_1^{(N+1)}, t_2^{(1)}, t_2^{(N+1)}, \dots, t_N^{(1)}, t_N^{(N+1)})$ を結合して副鍵

$$k_i = (t_1^{(1)}, t_1^{(N+1)}, t_2^{(1)}, t_2^{(N+1)}, \dots, t_N^{(1)}, t_N^{(N+1)})$$

$$(i=1, 2, \dots, N)$$

を出力するビット結合手段、

とを含む。

5 1. 請求項4 8に記載の装置において、上記H関数手段は、

上記各 L_j を各1ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(2N)}) = L_j \quad (j=1, 2, \dots, M)$$

に分割するビット分割手段と、

上記ビット $(t_j^{(1)}, t_j^{(2)}, \dots, t_j^{(2N)})$ から i に対して k_i のビット位置 q により決まる L_j のビット位置の情報が、 k_i のビット位置となるよう結合して副鍵

$$k_i = (t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \dots, t_N^{(i)}, t_N^{(N+i)}) \quad (i=1, 2, \dots, N)$$

を出力するビット結合手段、

とを含む。

5.2. 請求項4.7又は4.8に記載の装置において、

上記G関数手段は次の演算処理、

$(L_{j+1}, (Y_{j+1}, v_{j+1})) = G(Y_j, v_j)$ ($0 \leq j \leq M-1$)において、出力結果を

$$((Y_j^{(1)}, Y_j^{(2)}, Y_j^{(3)}), \cdot v_j) \rightarrow \\ \left((L_{j+1}^{(1)}, L_{j+1}^{(2)}, L_{j+1}^{(3)}, L_{j+1}^{(4)}), [(Y_{j+1}^{(1)}, Y_{j+1}^{(2)}, Y_{j+1}^{(3)}, Y_{j+1}^{(4)}), v_{j+1}] \right)$$

ここで、 $Y_{j+1}^{(i)} = f(Y_j^{(i)}) \quad (i=1, 2, 3, 4)$

$$L_{j+1}^{(0)} = v_j$$

$$L_{j+1}^{(i)} = f(L_{j+1}^{(i-1)}) \oplus Y_{j+1}^{(i)} \quad (i=1, 2, 3, 4)$$

$$v_{j+1} = L_{j+1}^{(4)}$$

を実行する手段であり、

上記H関数手段は次の演算処理

$$k_i = H(i, L_1, L_2, \dots, L_N)$$

において、

$$q_{4i+j} = L_{j+1}^{(i+1)} \quad (i=0, 1, 2, 3)$$

$$(t_i^{(0)}, t_i^{(1)}, \dots, t_i^{(7)}) = q_i \quad (i=0, 1, \dots, 31)$$

$$k_{(i+1)} = \left(t_{0+(i \bmod 2)}^{([i/2])}, t_{2+(i \bmod 2)}^{([i/2])}, \dots, t_{30+(i \bmod 2)}^{([i/2])} \right) \quad (i=0, 1, \dots, N-1)$$

を実行する手段である。

5 3. 主鍵Kを入力して複数の副鍵 k_i ($i = 1, \dots, N$) を生成する暗号鍵スケジュール装置をコンピュータにより実行させるためのプログラムを記録した記録媒体であって、

上記プログラムは、 Y_0 としての主鍵Kと定数 v_0 とを入力として入力を拡散処理することを継続的に複数回繰返し、各拡散処理ごとに中間鍵 L_j ($j=1, 2, \dots, M$) を出力する中間鍵生成処理と、

上記各生成された中間鍵 L_j を記憶部に記憶する処理と、

上記記憶部に所定数の中間鍵 $L_1 \sim L_M$ が記憶されると、副鍵 k_i の i によって決まる各ビット位置の $L_1 \sim L_M$ における情報を抽出して副鍵 k_i を生成する副鍵生成処理と、

を上記コンピュータにより実行させることを特徴とする記録媒体。

5 4. 主鍵Kを入力して複数の副鍵 k_i ($i=1, 2, \dots, N$) を生成する暗号鍵スケジュール装置をコンピュータにより実行させるためのプログラムを記録した記録媒体であって、

上記プログラムは、

Y_0 としての主鍵Kと定数 v_0 とを入力として、入力を拡散処理することを継続的に複数回繰返し、各拡散処理ごとに中間鍵 L_j ($j=1, 2, \dots, M$) を出力する中間鍵生成処理と、

上記各中間鍵 L_j が生成されるごとに、副鍵 k_i の i と、 k_i のビット位置 q により決まる L_j のビット位置の情報を k_i のビット位置の情報として抽出して、記憶部に記憶する処理と、

上記記憶部内の各副鍵 k_i の各ビット位置の情報が決まると、その副鍵 k_i を出力する処理と、

を処理コンピュータにより実行させることを特徴とする記録媒体。

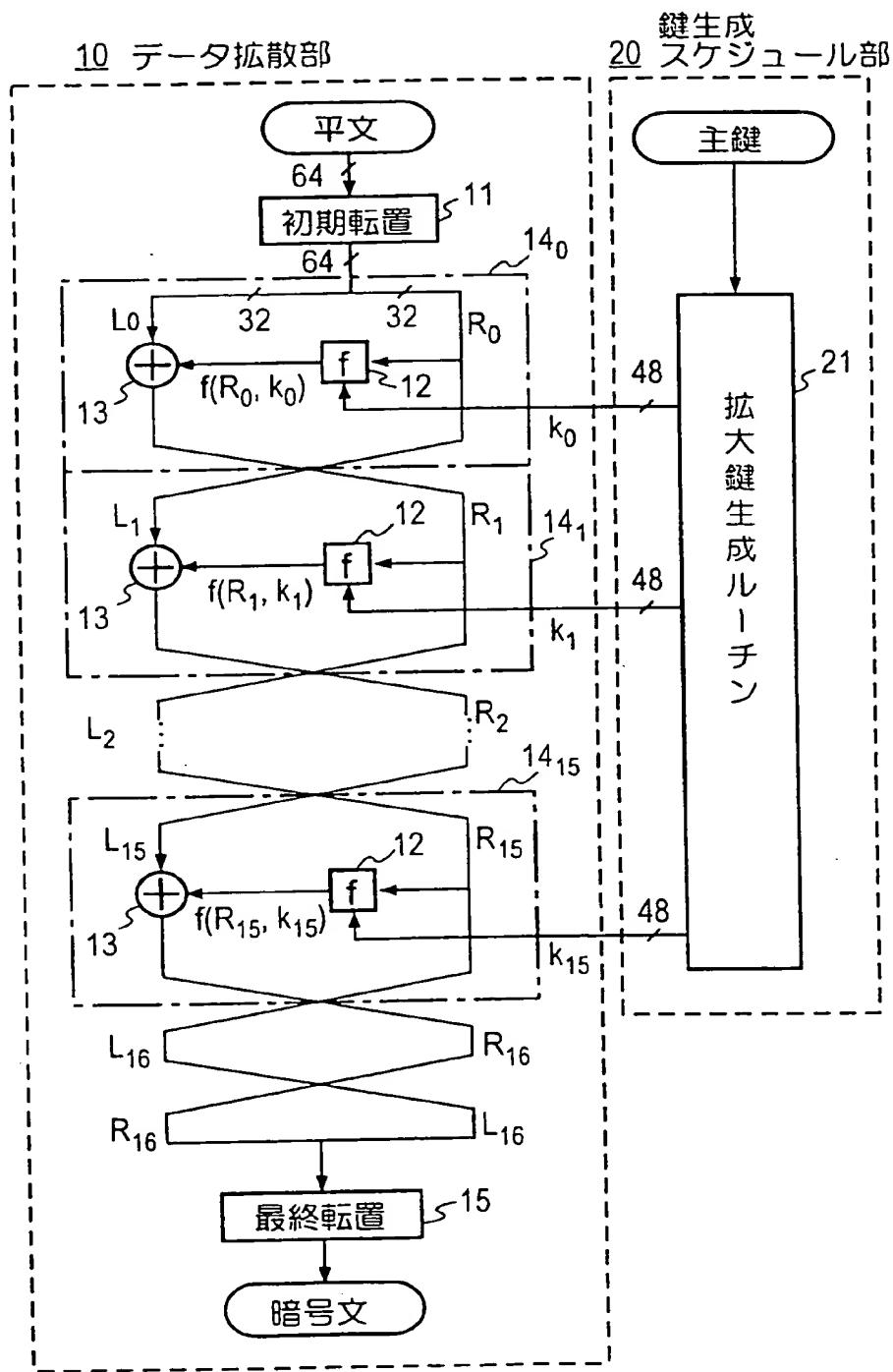


図1

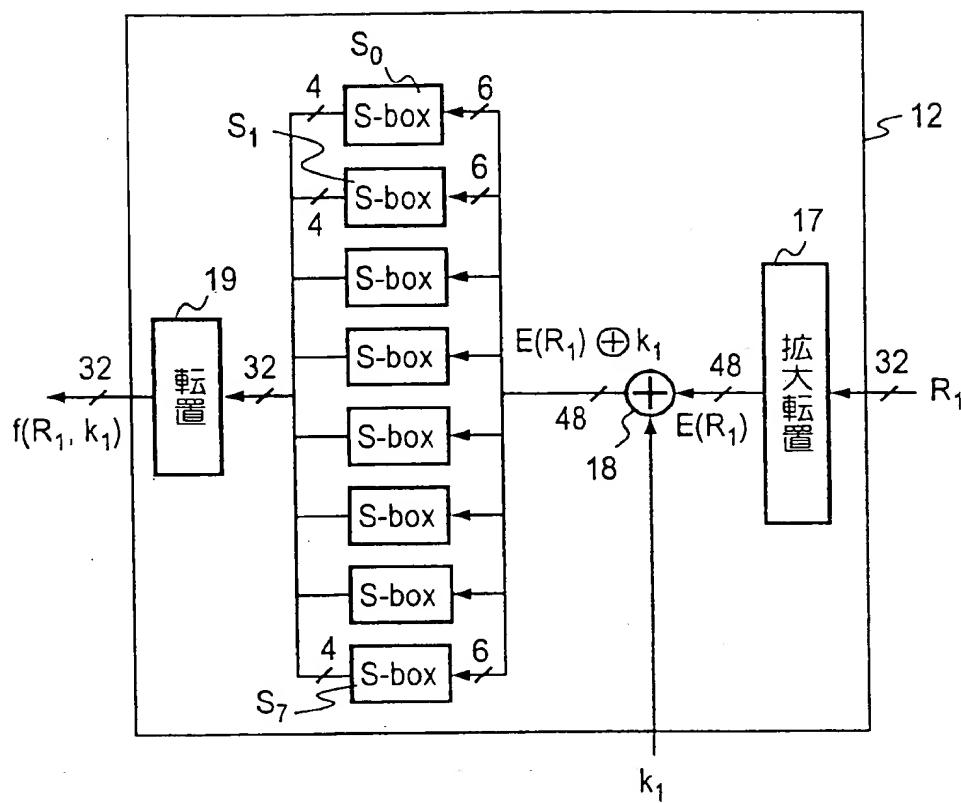
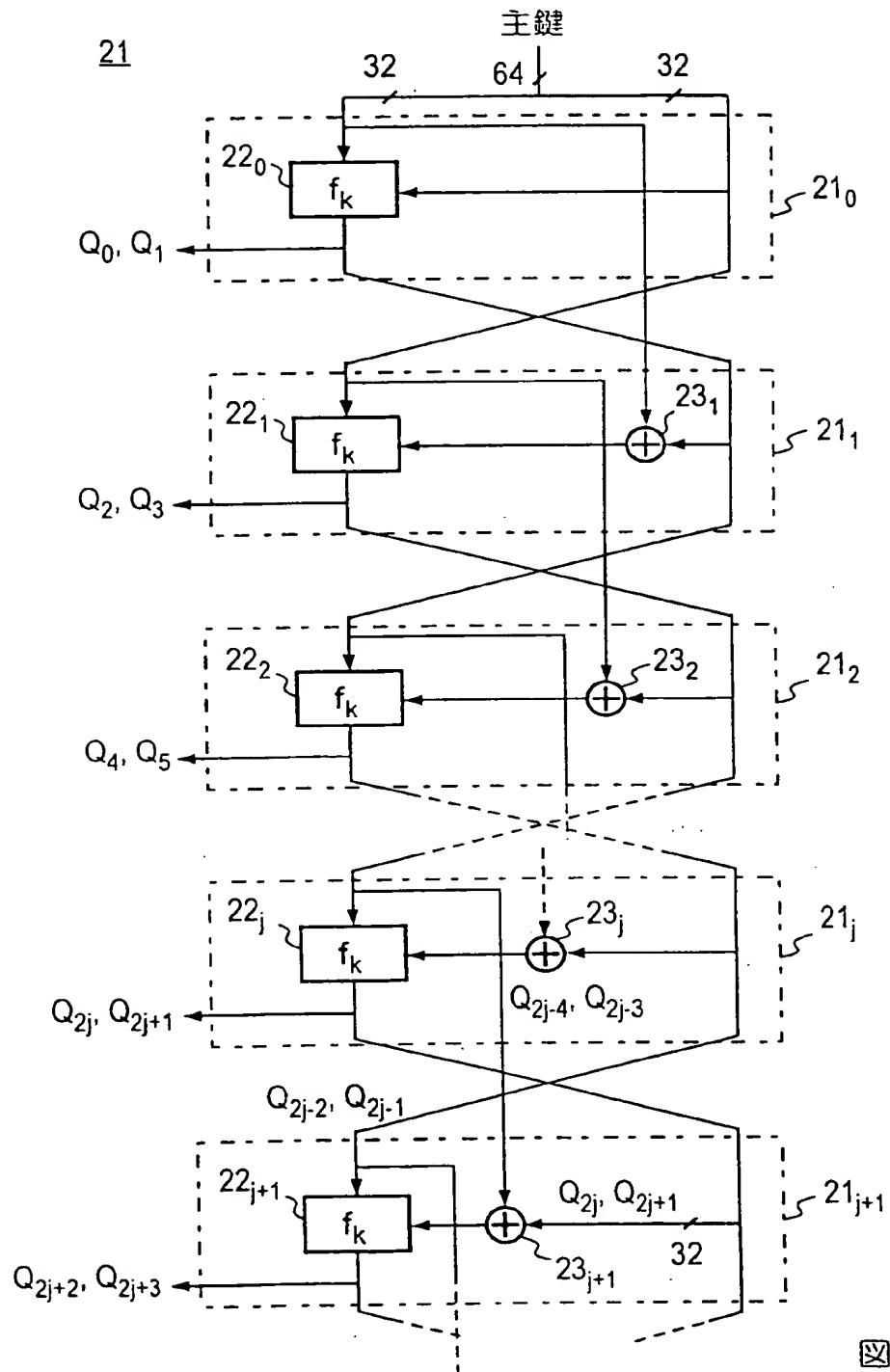


図2



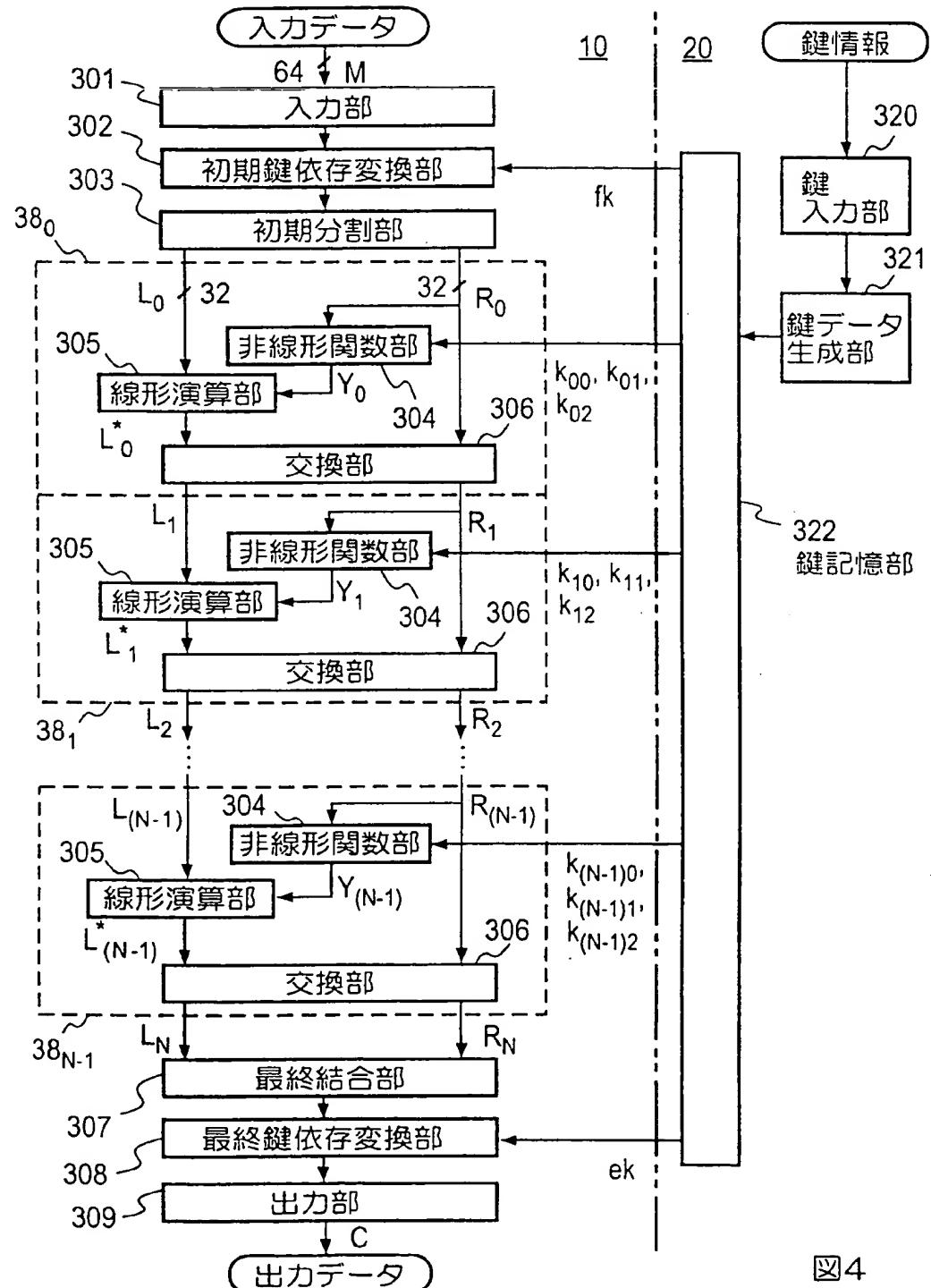


図4

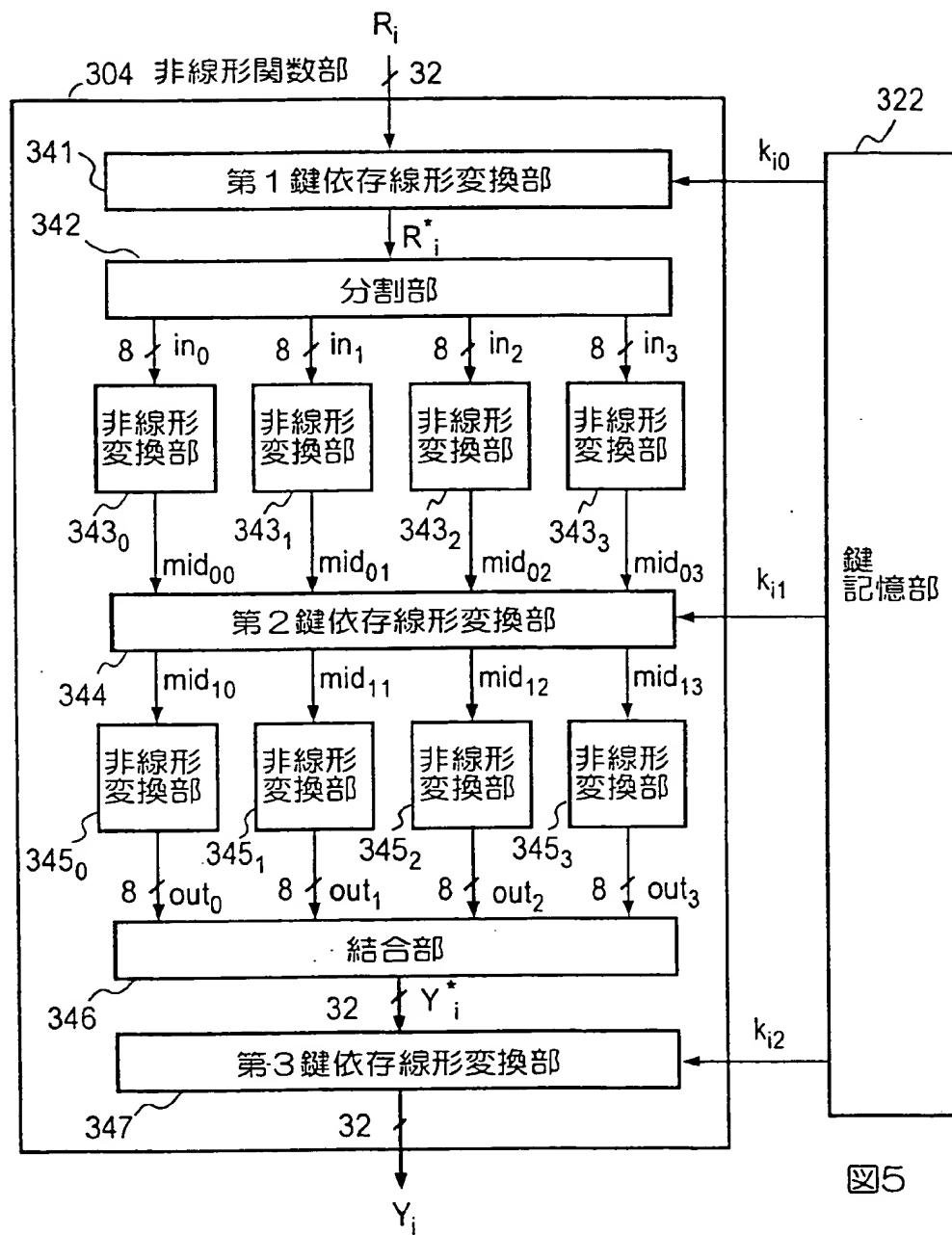


図5

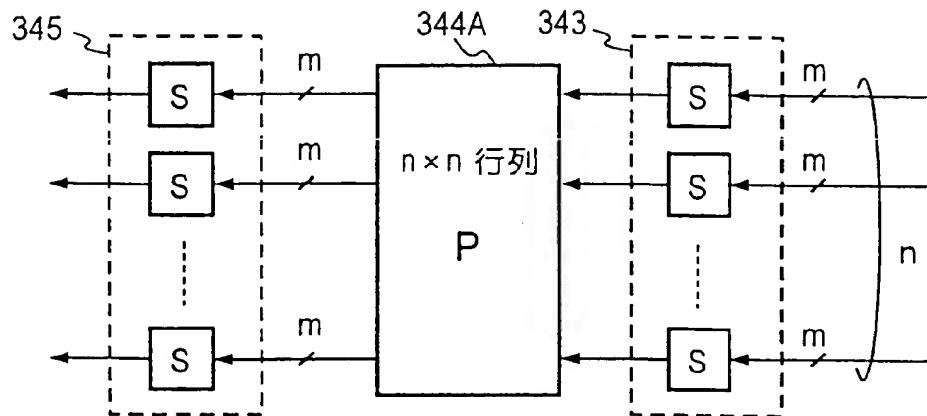


図6

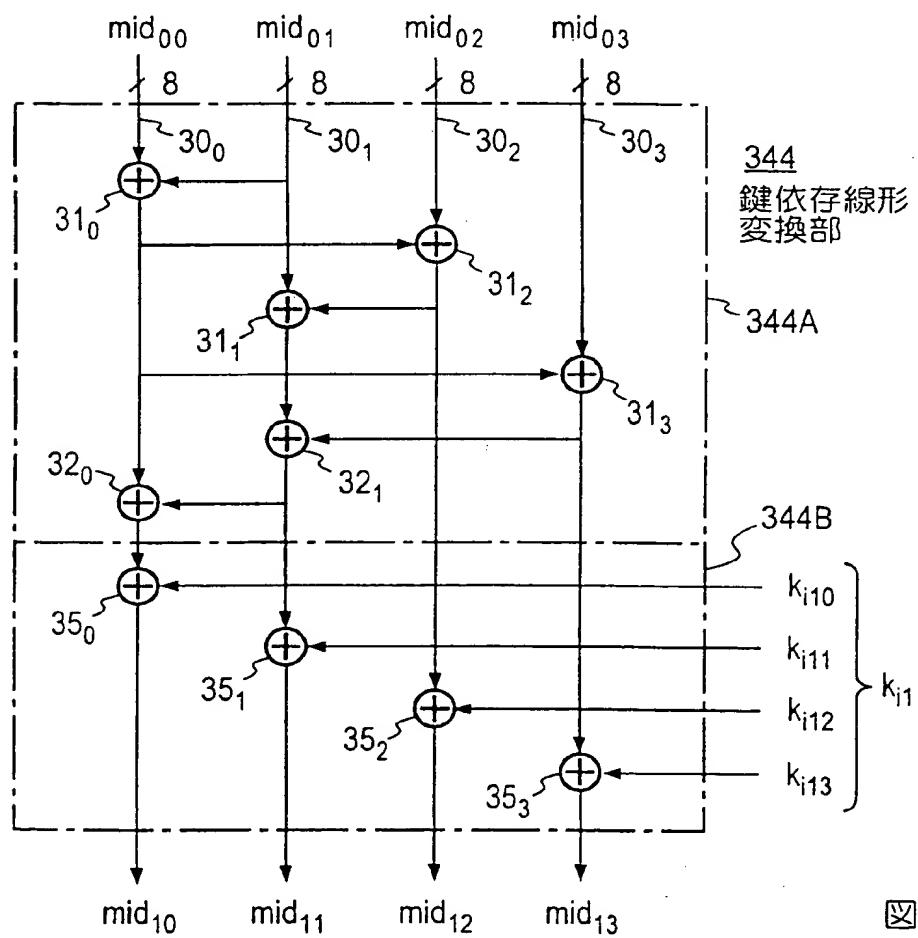
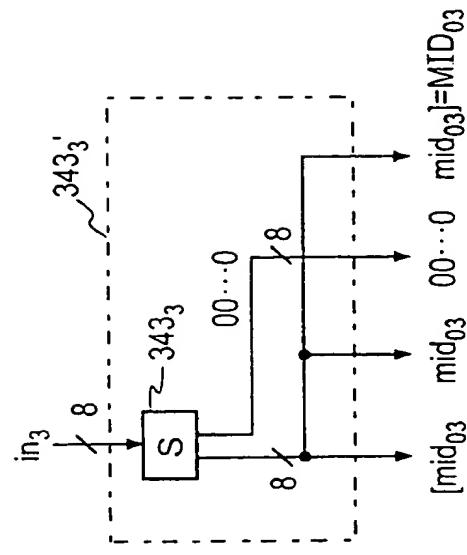
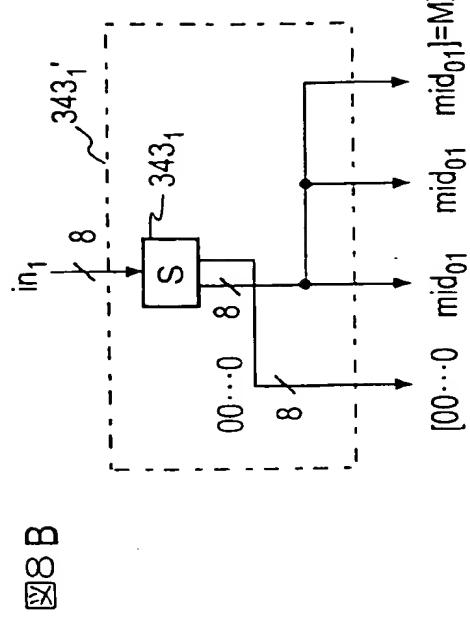
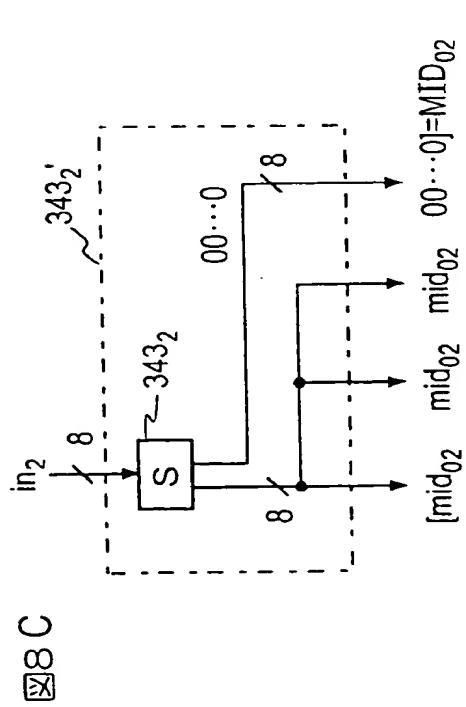
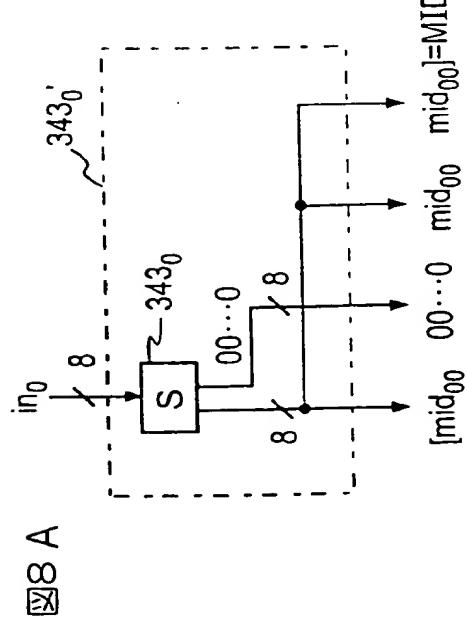


図7



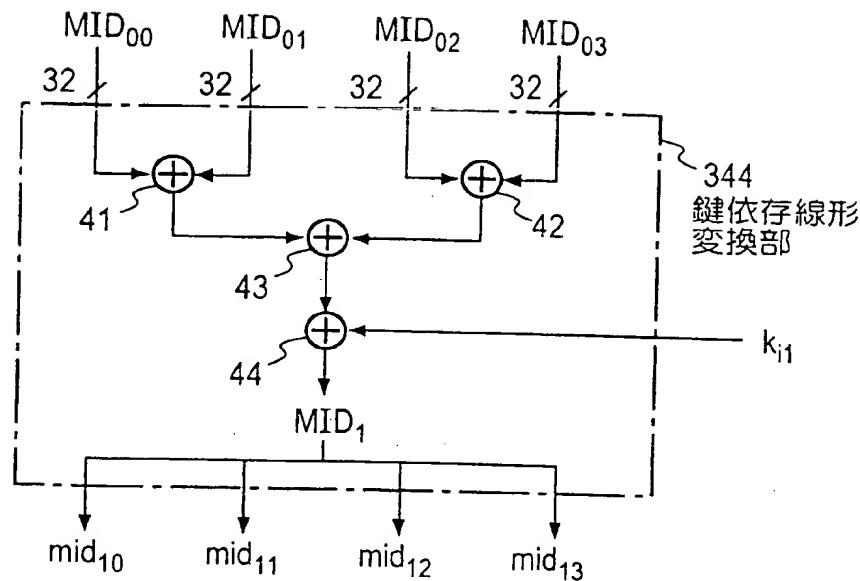


図9

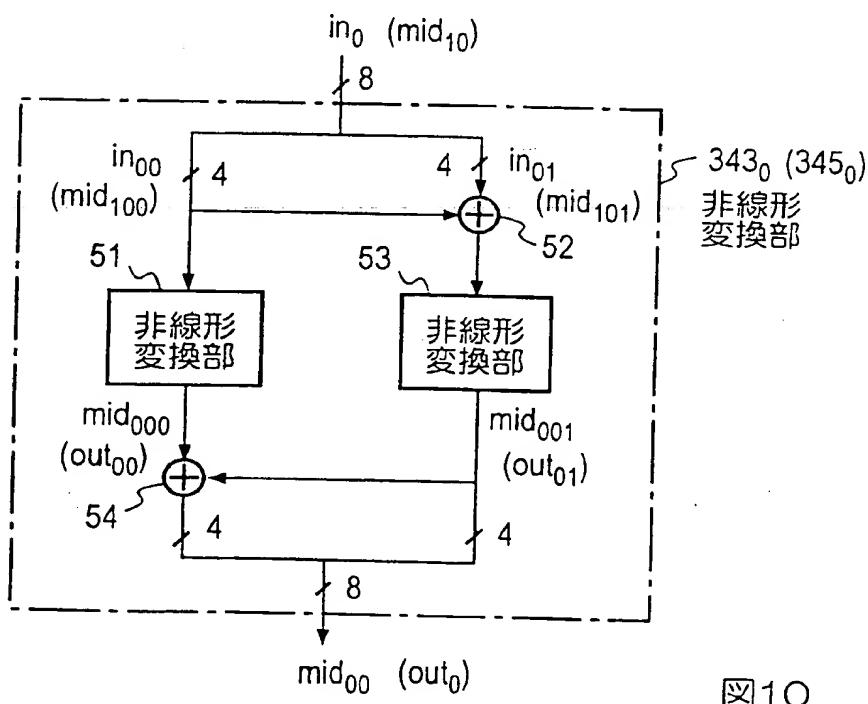


図10

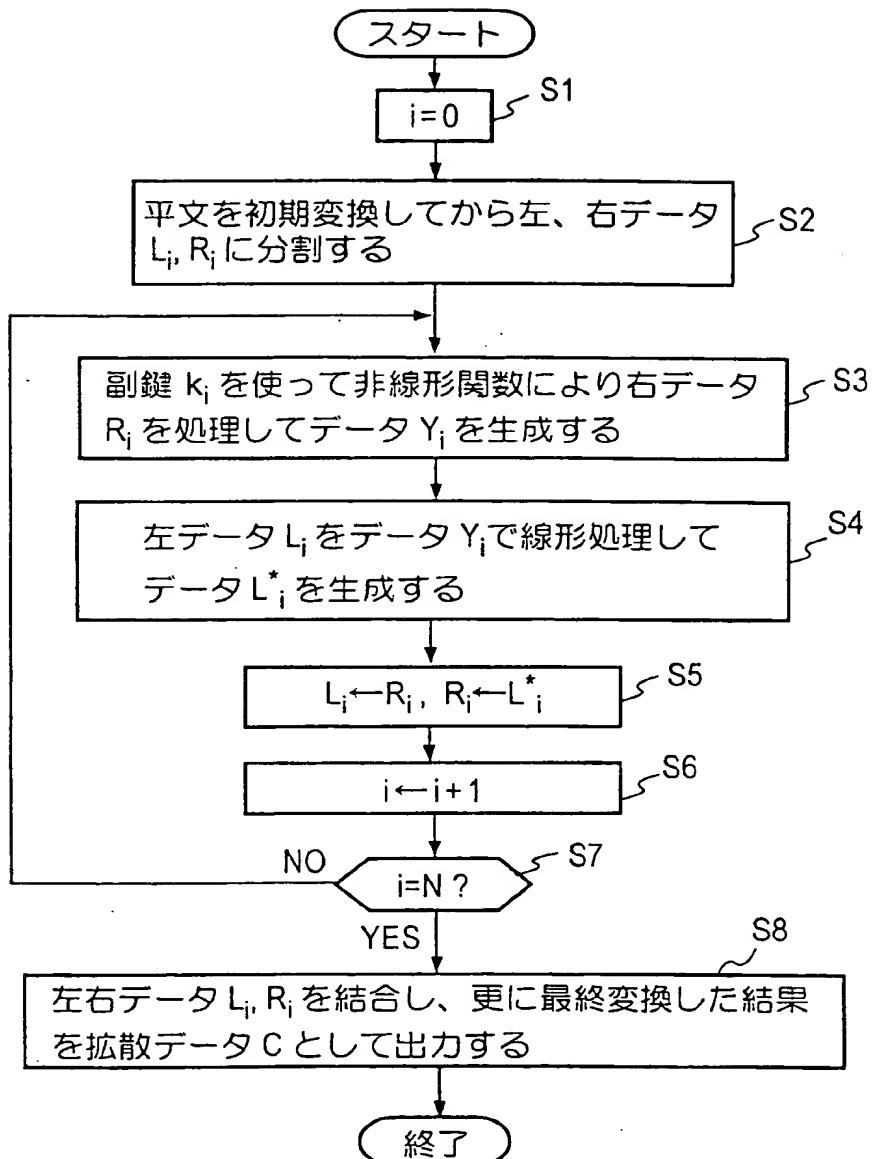


図11

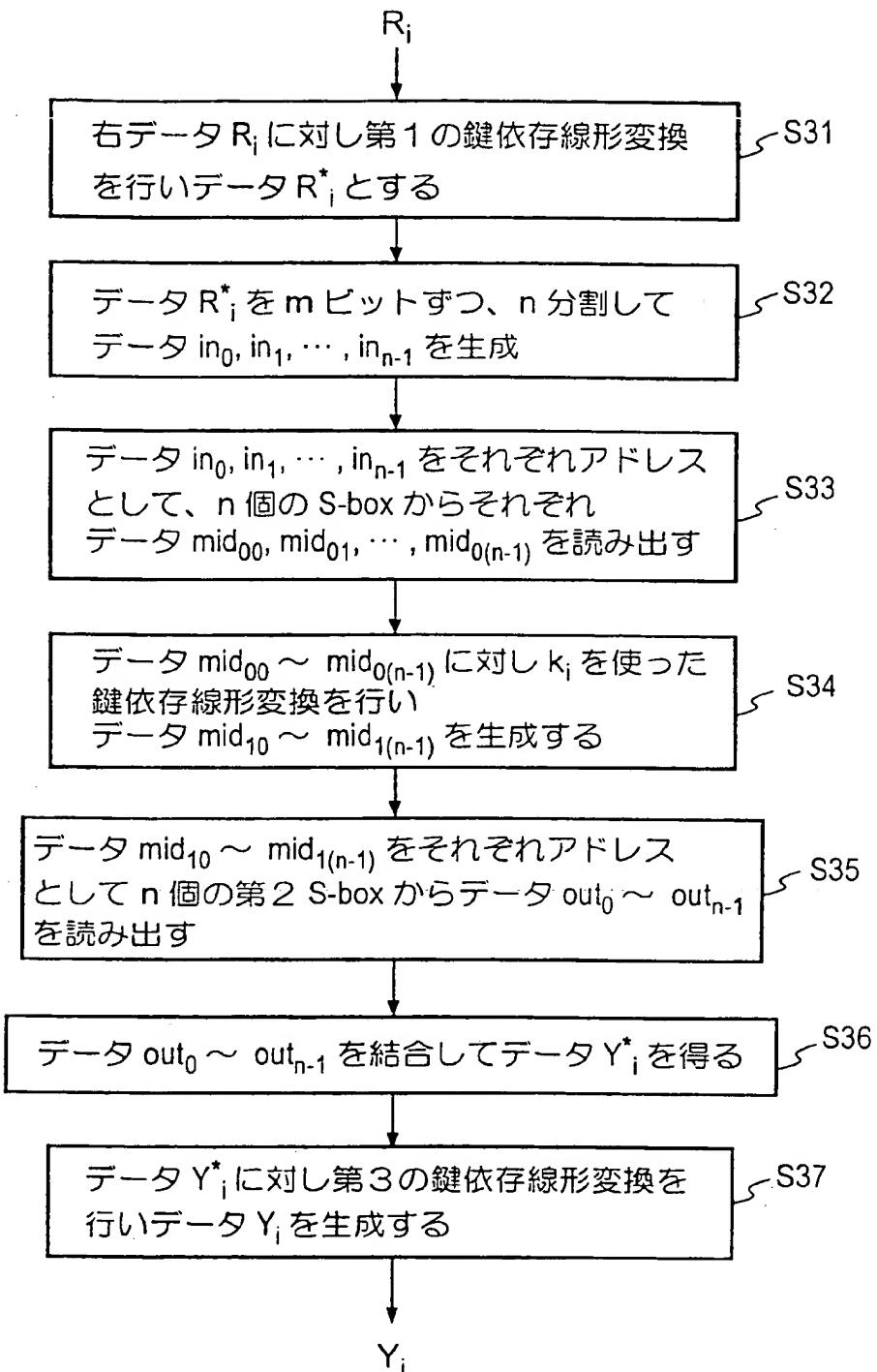


図12

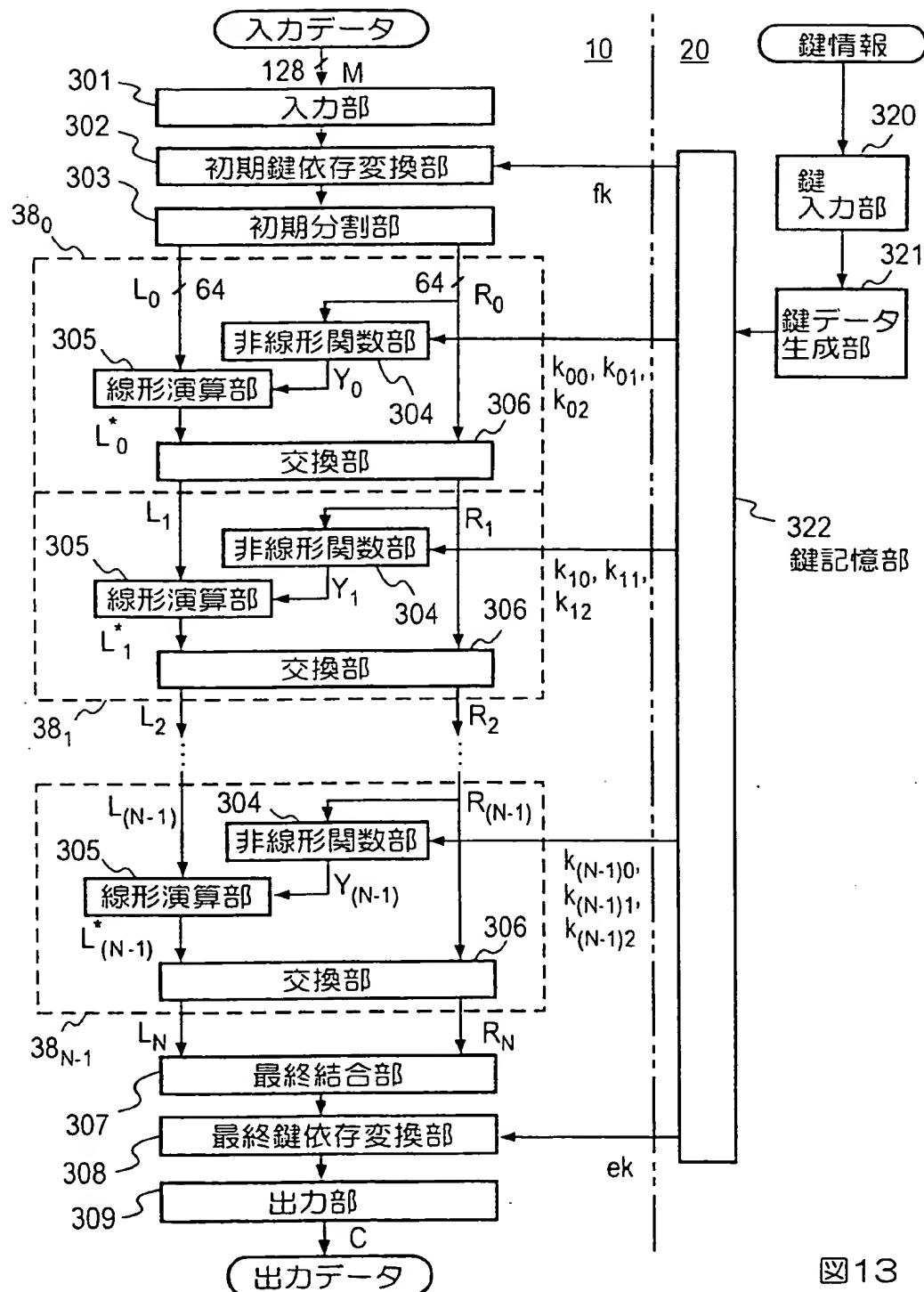


図13

図15A

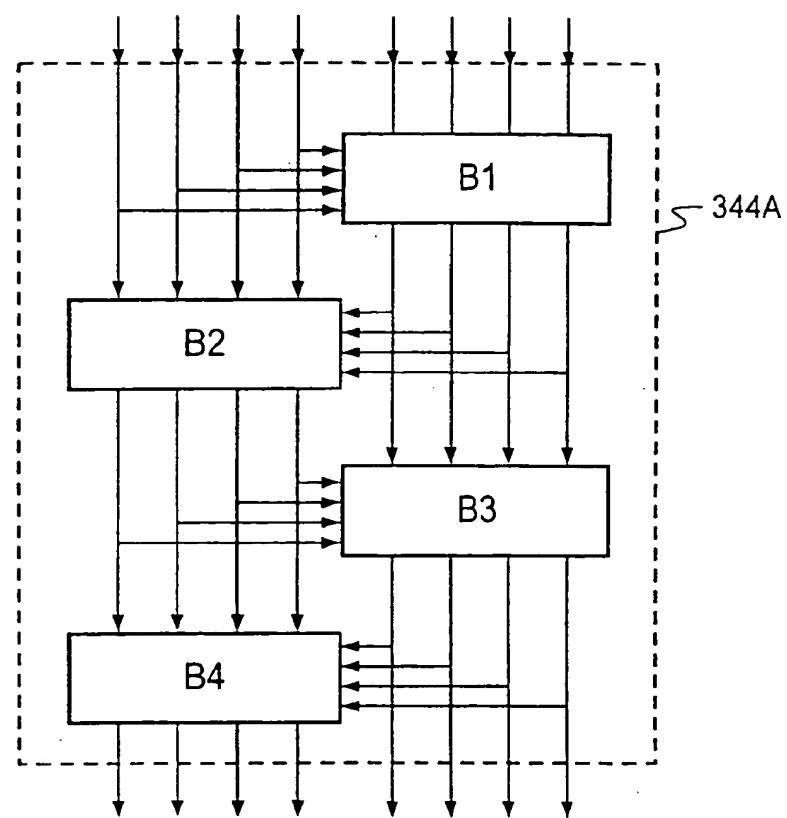
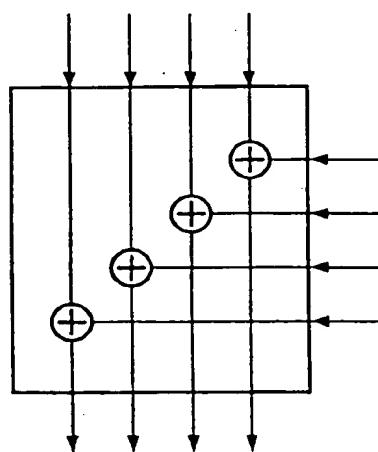


図15B



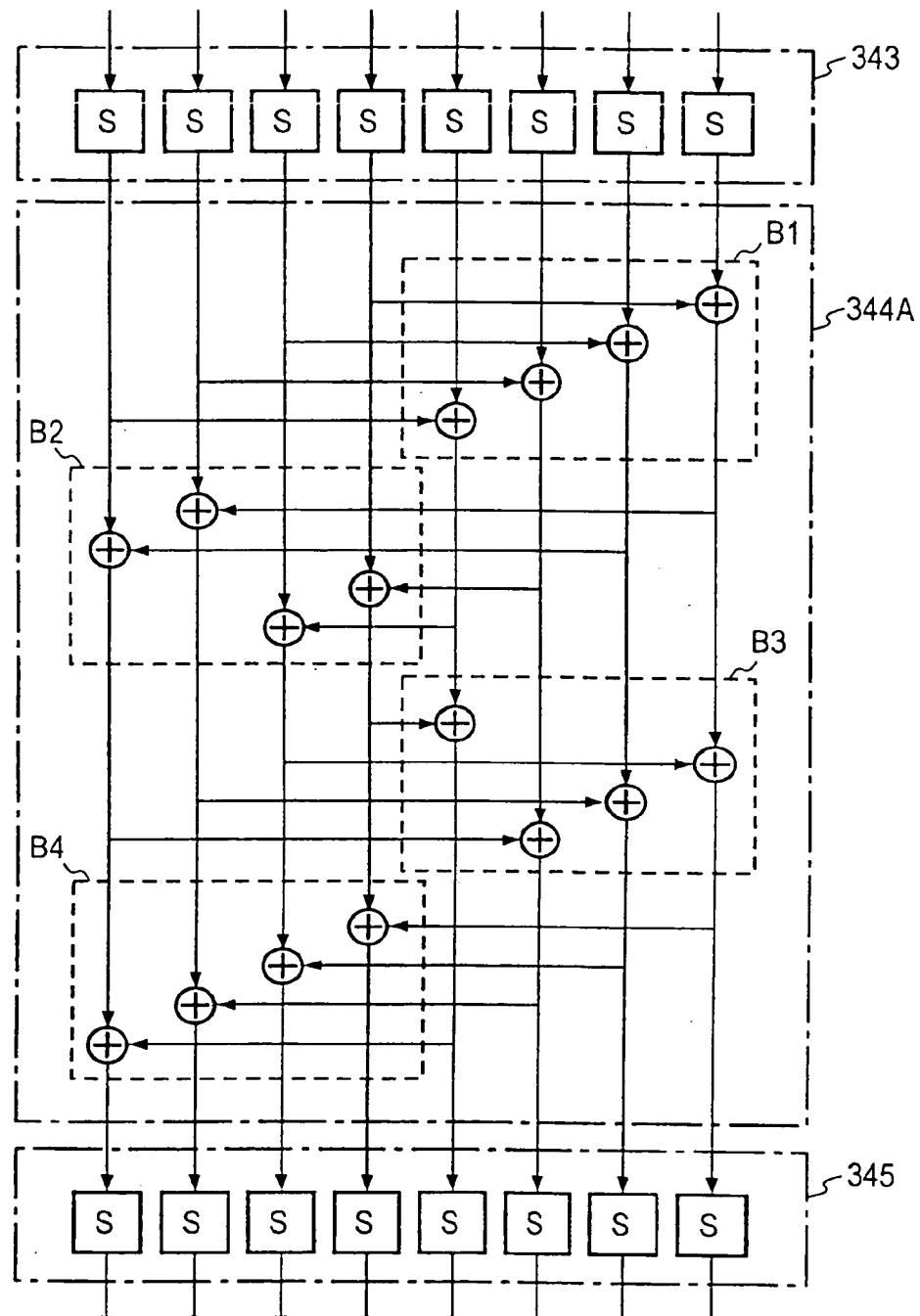


図16

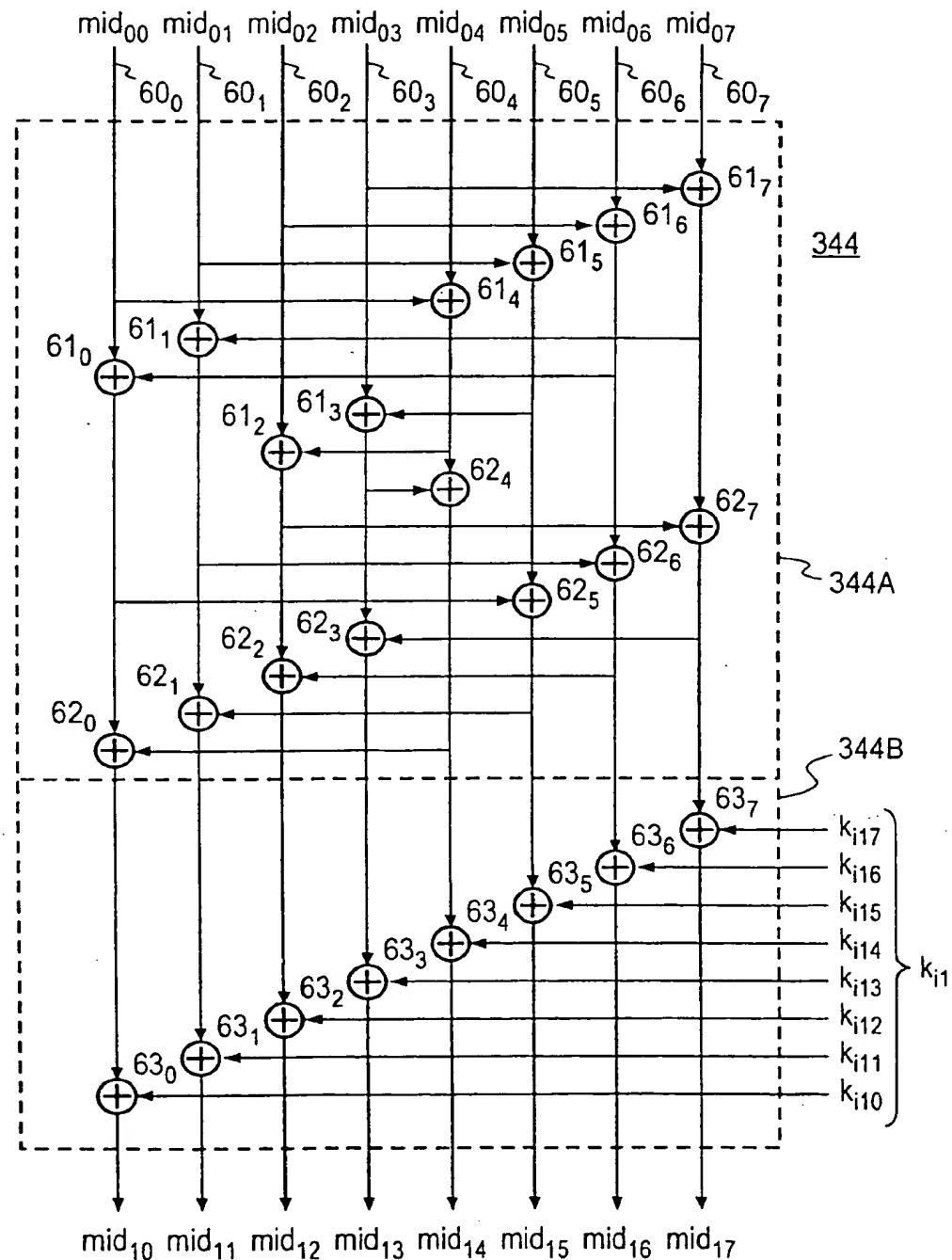
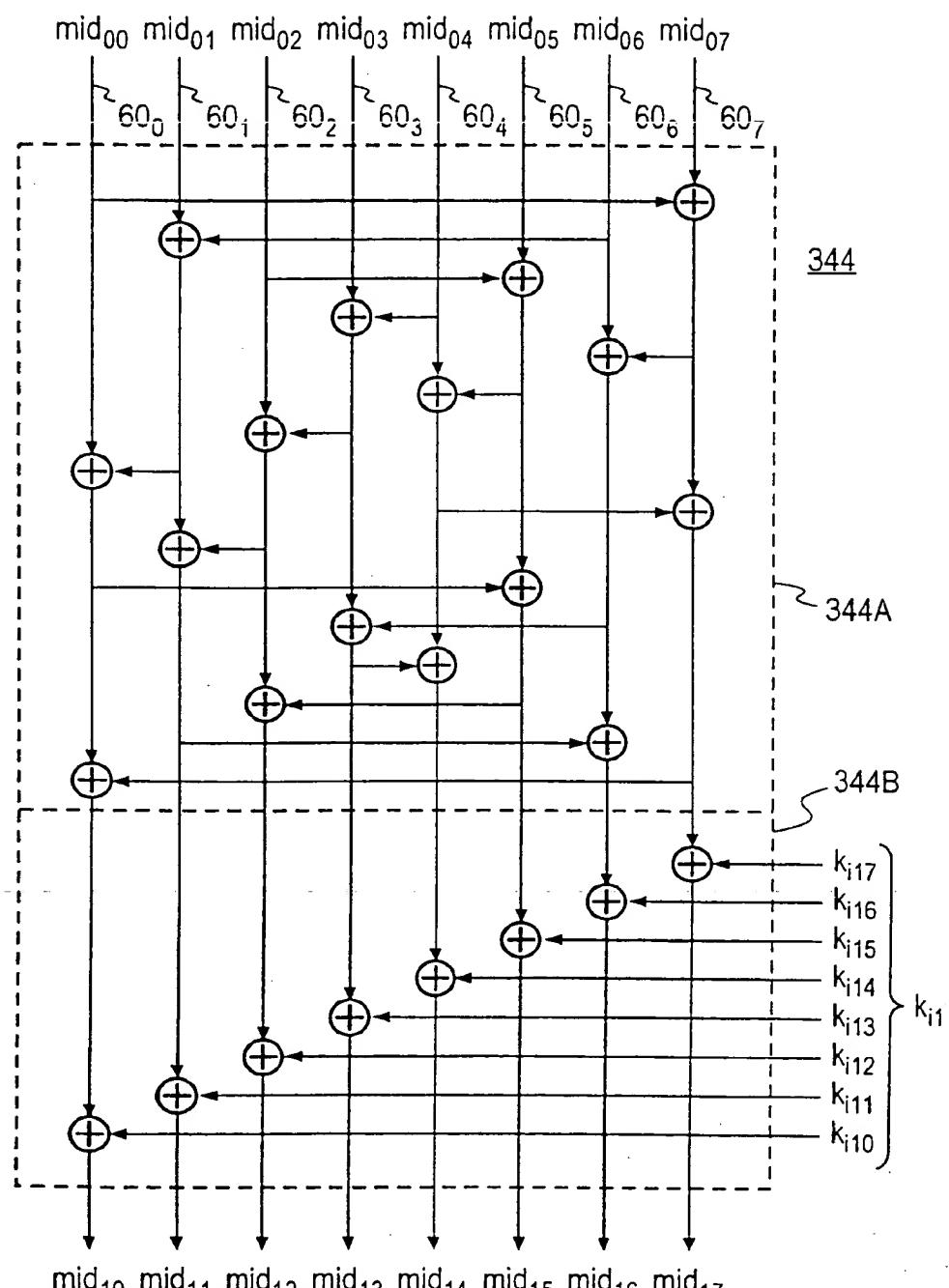
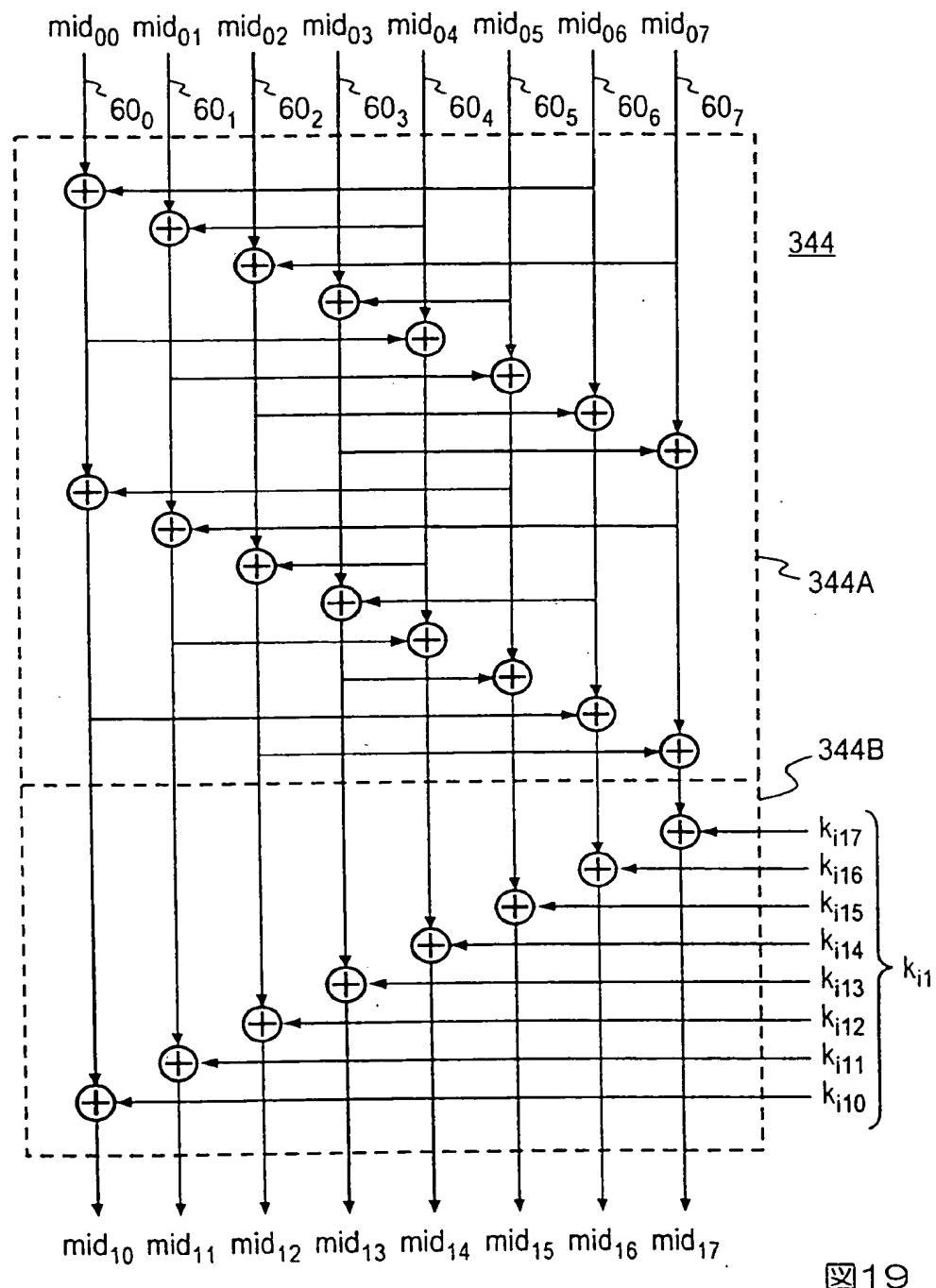
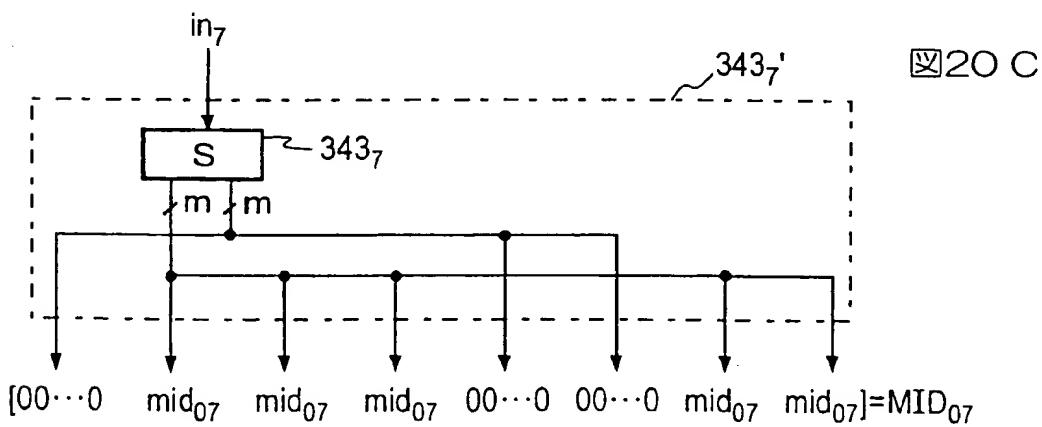
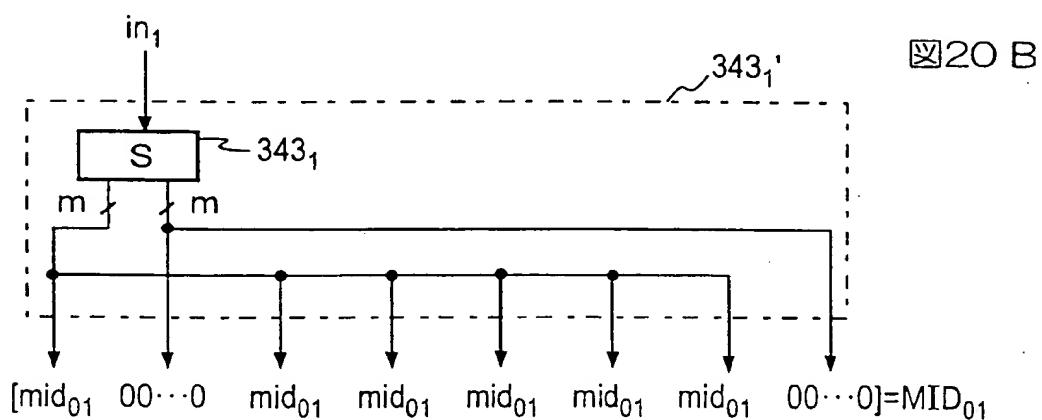
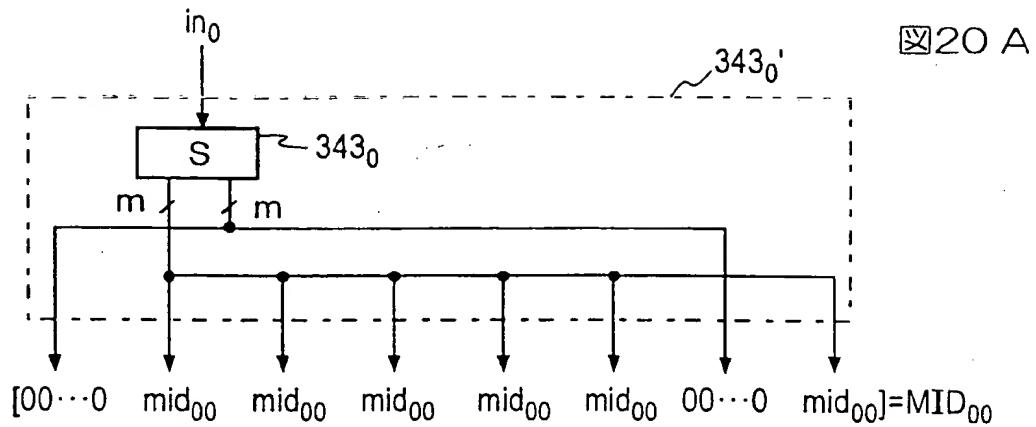


図17



18





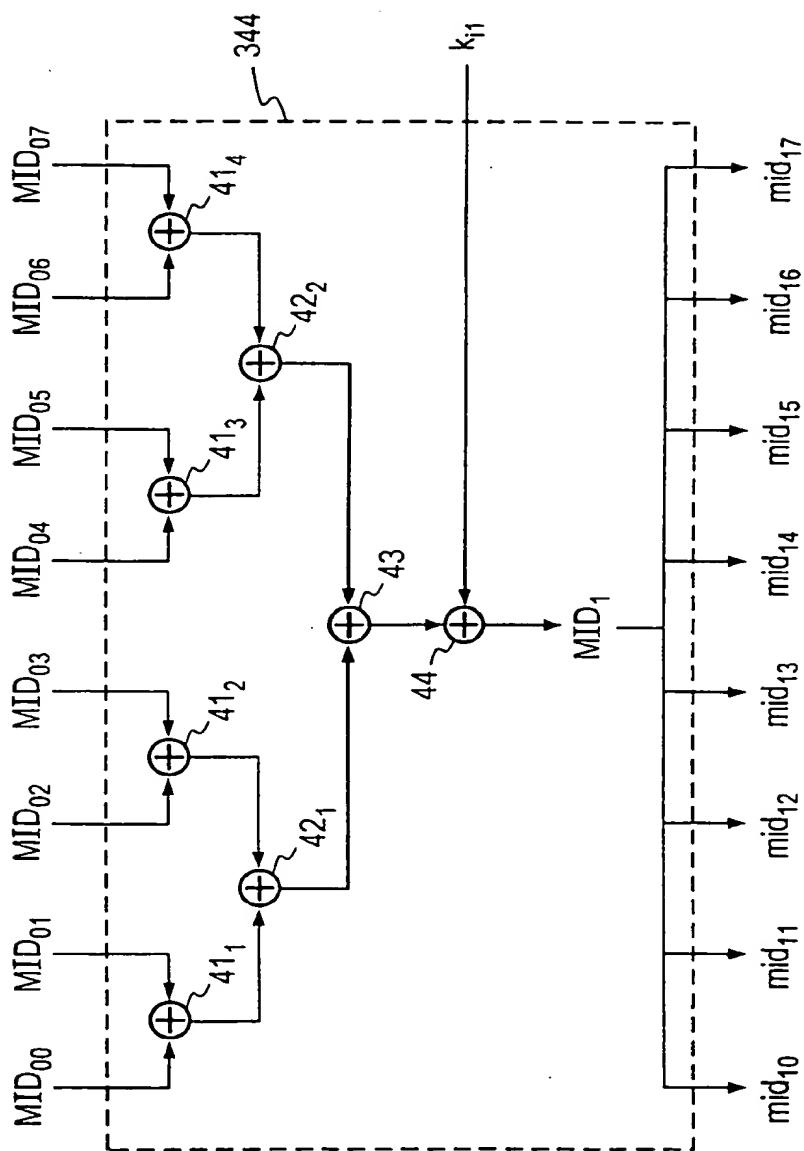


图21

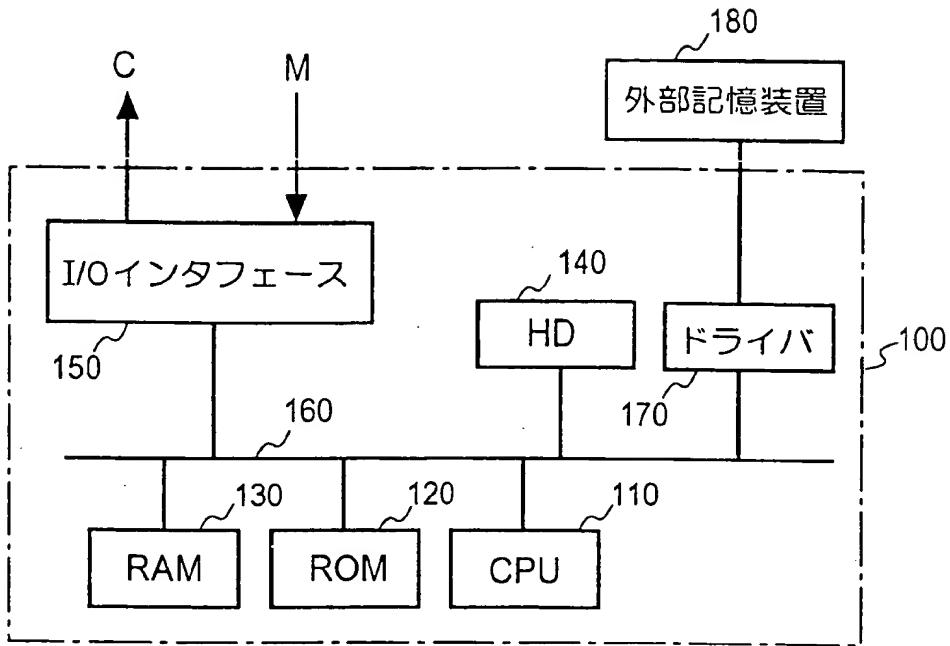
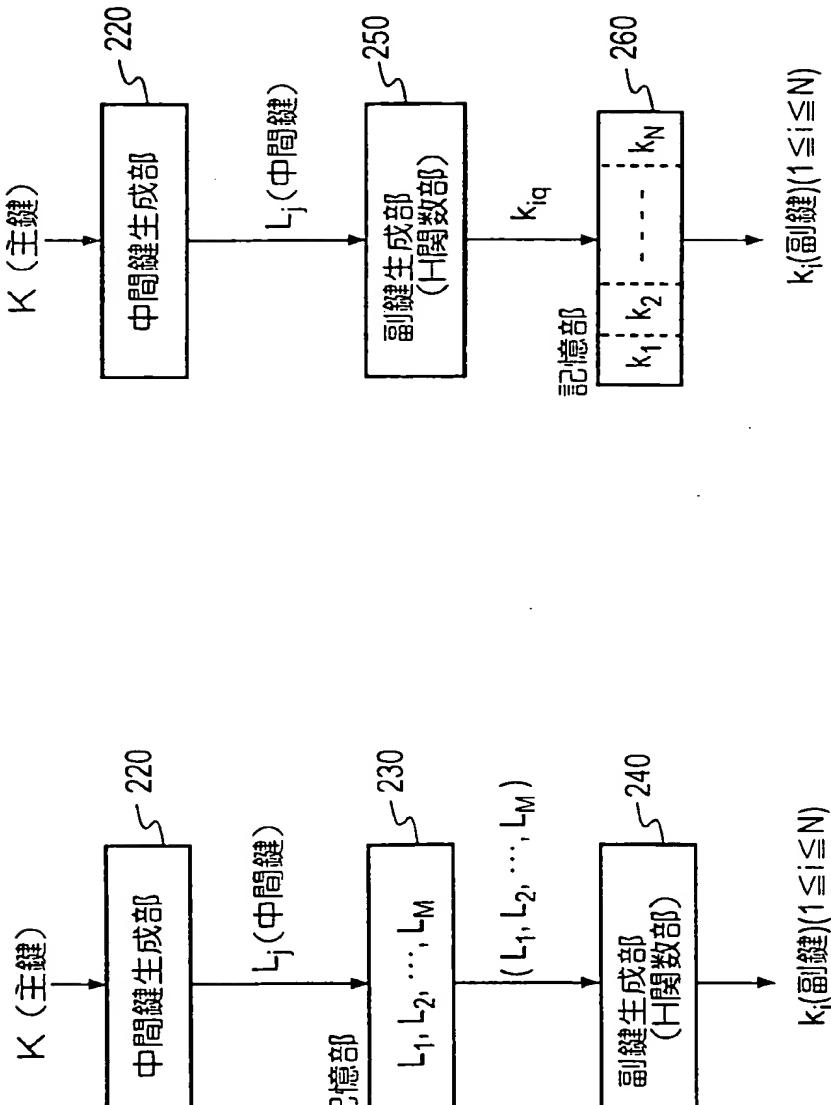


図22

図23A
図23B

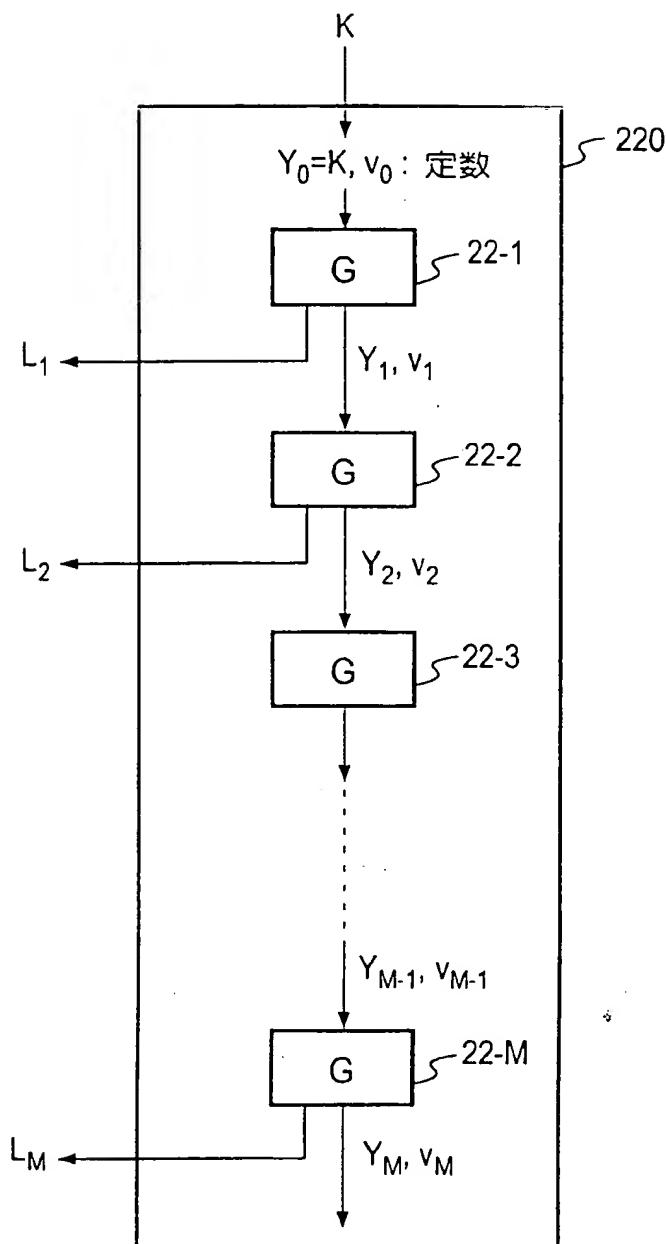


図24

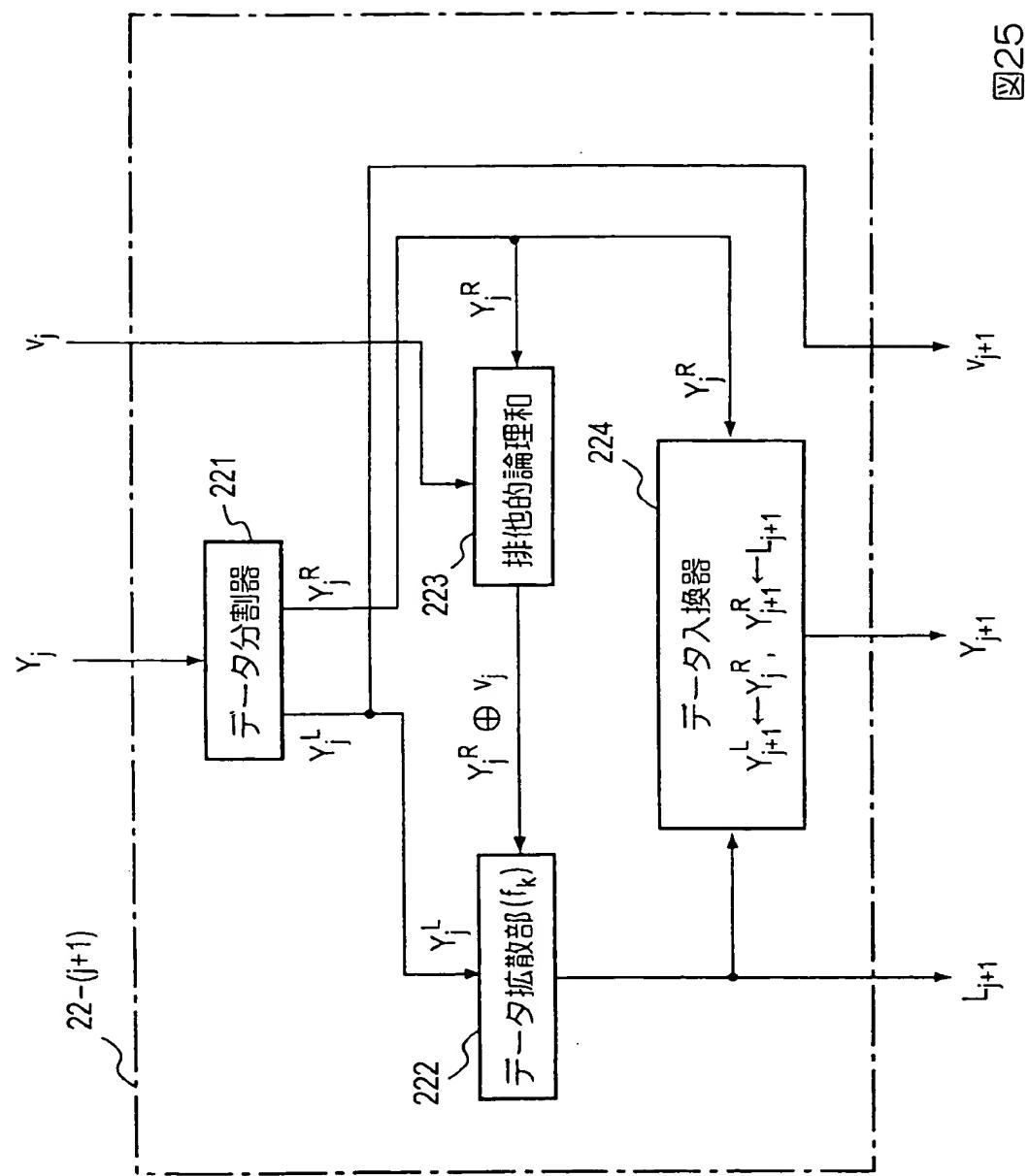


図25

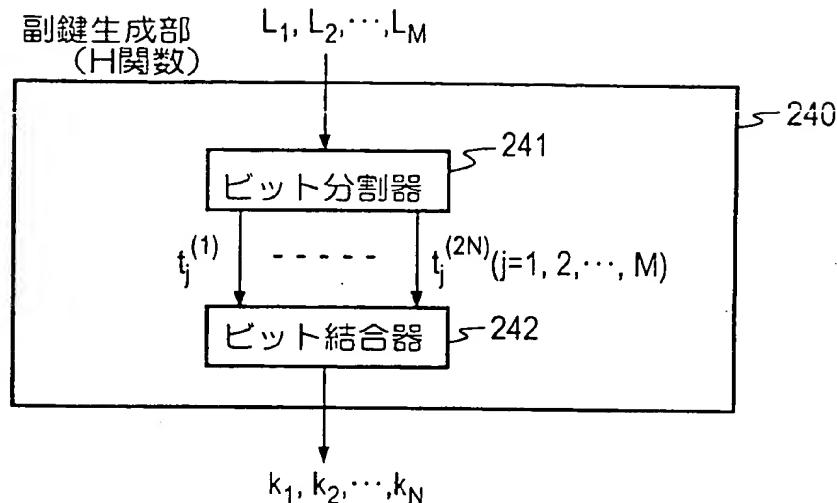


図26

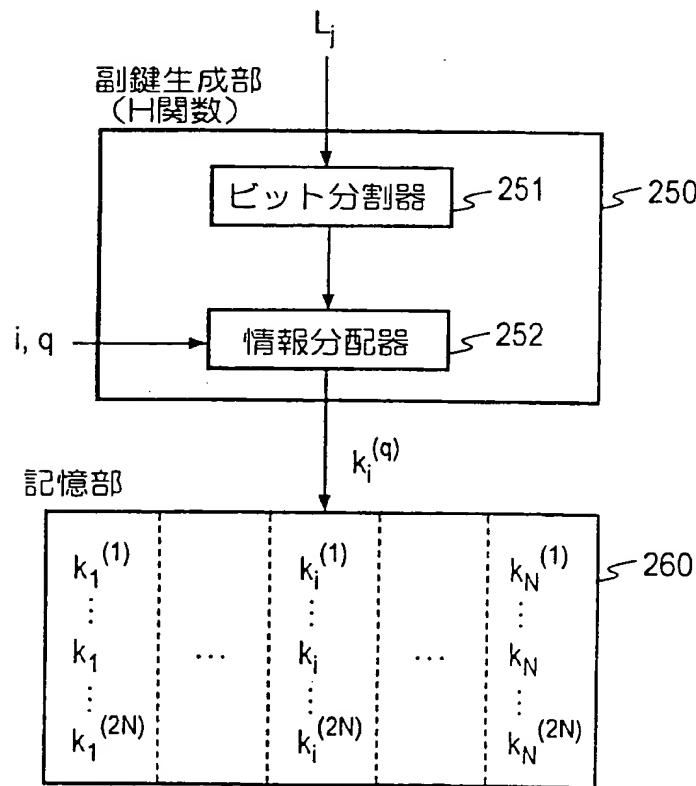


図27

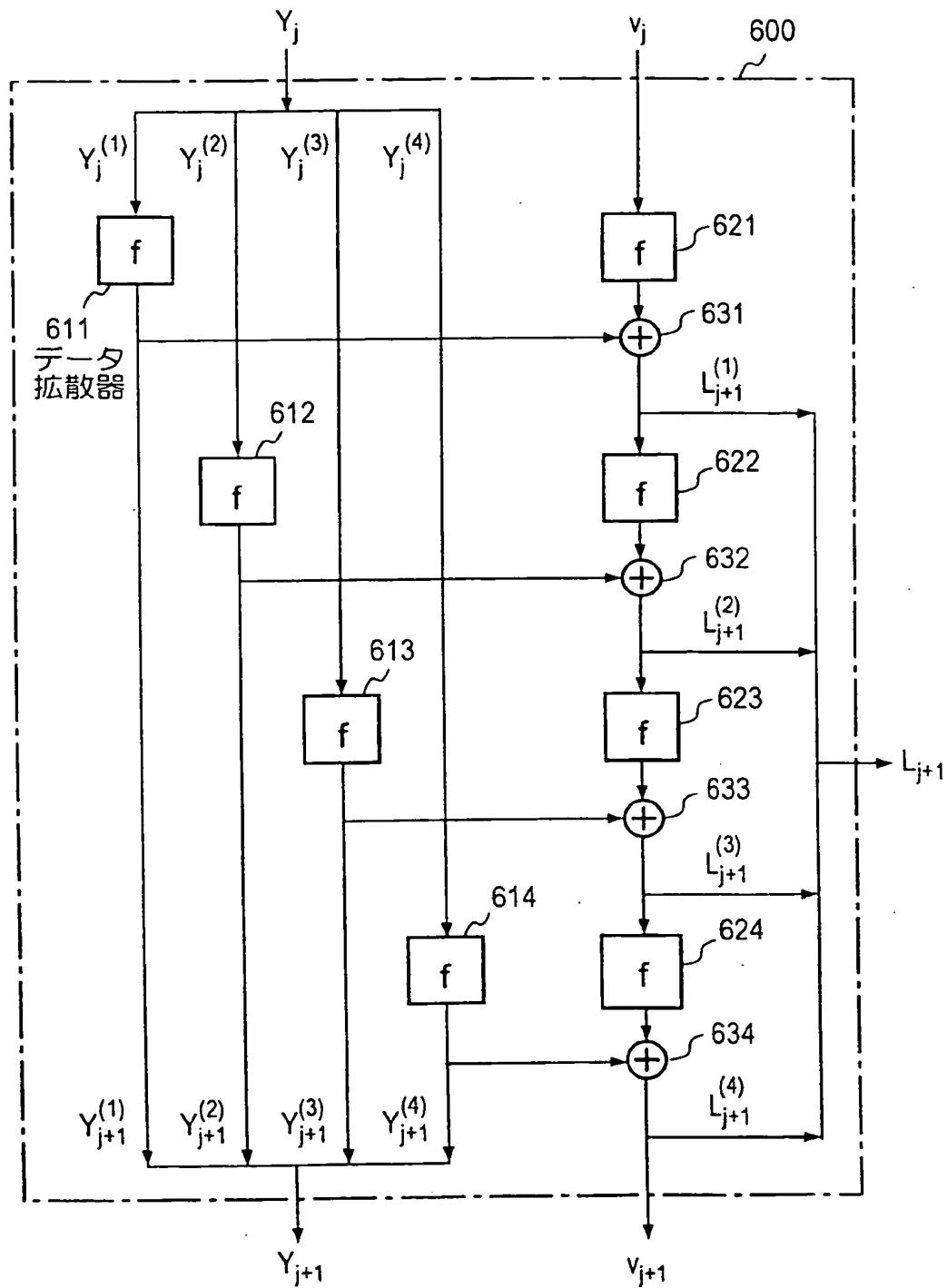


図28

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP99/00337

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁶ G09C1/00, H04L9/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl⁶ G09C1/00-5/00, H04K1/00-3/00, H04L9/00-9/38

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
JICST File (JOIS), INSPEC (DIALOG)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	M. Kanda, S. Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta and T. Matsumoto, "A New 128-bit Block Cipher E2," Technical Report of IEICE, Vol. 98, No. 227, (30 Sep 1998), p.13-24 (ISEC98-12) (in Japanese)	1-54
PA	M. Kanda, Y. Takashima and T. Matsumoto, "A round function structure consisting of few S-boxes (Part II)," The 1998 Symposium on Cryptography and Information Security, (28 Jan 1998), 2.2.D, (in Japanese)	1-46
PA	M. Kanda, Y. Takashima, T. Matsumoto, K. Aoki and K. Ohta, "A round function structure consisting of few s-boxes (Part III)," Technical Report of IEICE, Vol. 98, No. 48, (15 May 1998), p.21-30 (ISEC98-3) (in Japanese)	1-46
A	V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers and E. De Win, "The Cipher SHARK," Lecture Notes in Computer Science, Vol. 1039, (1996), p.99-111	1-54

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:	
"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier document but published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	"&" document member of the same patent family

Date of the actual completion of the international search
15 April, 1999 (15. 04. 99)

Date of mailing of the international search report
27 April, 1999 (27. 04. 99)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP99/00337

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	M. Matsui, "New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis," Lecture Notes in Computer Science, Vol. 1039, (1996), p.205-218	1-54
A	A. Shimizu and S. Miyaguchi, "Fast Data Encipherment Algorithm FEAL," Lecture Notes in Computer Science, Vol. 304, (1987), p.267-278	1-54

国際調査報告

国際出願番号 PCT/JP99/00337

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl' G 09 C 1/00
H 04 L 9/06

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl'	G 09 C	1/00	-	5/00
	H 04 K	1/00	-	3/00
	H 04 L	9/00	-	9/38

最小限資料以外の資料で調査を行った分野に含まれるもの

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

J I C S T ファイル (J O I S)
I N S P E C (D I A L O G)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
P X	M. Kanda, S. Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta and T. Matsumoto, "A New 128-bit Block Cipher E2," Technical Report of IEICE, Vol. 98, No. 227, (30 Sep 1998), p. 13-24 (ISEC98-12) (in Japanese)	1-54
P A	M. Kanda, Y. Takashima and T. Matsumoto, "A round function structure consisting of few S-boxes (Part II)," The 1998 Symposium on Cryptography and Information Security, (28 Jan 1998), 2.2.D, (in Japanese)	1-46

 C欄の続きにも文献が列挙されている。 パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「I」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日 15. 04. 99	国際調査報告の発送日 27.04.99
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 郵便番号 100-8915 東京都千代田区霞が関三丁目4番3号	特許庁審査官 (権限のある職員) 丸山 高政 5W 9570 電話番号 03-3581-1101 内線 6447

C (続き) 関連すると認められる文献		関連する 請求の範囲の番号
引用文献の カテゴリーや*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	
P A	M. Kanda, Y. Takashima, T. Matsumoto, K. Aoki and K. Ohta, "A round function structure consisting of few s-boxes (Part III)," Technical Report of IEICE, Vol. 98, No. 48, (15 May 1998), p. 21-30 (ISEC98-3) (in Japanese)	1-46
A	V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers and E. De Win, "The Cipher SHARK," Lecture Notes in Computer Science, Vol. 1039, (1996), p. 99-111	1-54
A	M. Matsui, "New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis," Lecture Notes in Computer Science, Vol. 1039, (1996), p. 205-218	1-54
A	A. Shimizu and S. Miyaguchi, "Fast Data Encipherment Algorithm FEAL," Lecture Notes in Computer Science, Vol. 304, (1987), p. 267-278	1-54

THIS PAGE BLANK (USPTO)